Universidad Politécnica de Valencia

Departamento de Comunicaciones

Tesis Doctoral

« Modelado y Evaluación de la Gestión de Recursos en Redes Móviles Celulares »

Autor: Mª José Domenech Benlloch
Ing. de Telecomunicación
Director: Vicente Casares Giner
Dr. Ing. de Telecomunicación

Valencia
Abril 2009

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

"Little Gidding"
T. S. Eliot

Agradecimientos

Con la escritura de este trabajo culmina, de algún modo, una larga etapa de formación académica comenzada en las aulas de pre-escolar del C.P. José Senent. A través de estas líneas me gustaría agradecer a todos aquellos que me han ofrecido su conocimiento y su trabajo para que yo pudiera dar el siguiente paso de esta larga andadura y también a aquellos que me han acompañado y dado fuerzas y ánimo para seguir adelante —*mamà, papà, Da, gràcies*—.

Obviamente, mencionar a todos a los que va dirigido este agradecimiento es casi imposible. Sin embargo, me gustaría mencionar expresamente a la gente cuya relación con este trabajo es más directa: a Vicente, por su dirección y ayuda en la realización de esta tesis, pero también al resto de miembros del grupo GIRBA —Pablo, José Ramón, Miguel Ángel, Luís— por estar ahí y, en especial, a Jorge y Vicent por su asistencia sin la cual este trabajo no hubiera sido posible. También me gustaría destacar a los miembros de Laboratorio con los que he coincidido a lo largo de los años: a Nacho, Ángel y Elena, por haber hecho del *Lab* algo más que un lugar de trabajo; asimismo, destacar a José Manuel y a David con los que he compartido largas horas de trabajo, gracias por vuestra paciencia, por vuestra dedicación y por vuestra amistad.

Abstract

In the last decade, mobile networks have experienced an enormous growth. Moreover, due to the increase in the number of services and in the required bandwidth, it is necessary to develop accurate models and resource management mechanisms. This work aims to be a contribution to the development of models for the study and evaluation of radio resource management in mobile cellular networks. More specifically, the aim has been directed to the study retrial models. These models are useful for the characterization of different aspects of the network operation, such as modelling human behaviour and the characterization of new services offered by communication networks. Traditionally, a retrial system has been defined as a system in which users that are blocked, seek for access after a timeout. This characteristic is typical of the human behaviour and therefore, it should not be ignored when modelling a communication system, since it can have a great impact on the performance parameters of the system —blocking probability, probability forced termination ...—. Additionally, in mobile cellular networks, this efect can be found in the handovers due to the network structure and its characteristics. Thus, according to the GSM standard, for example, while the mobile is in the handover area —overlapped-area between the coverage of two or more cells—, and without the user's perception, it can ask the destination cell for resources a limited number of times. A correct characterization of this retrial process improves the performance of the network, avoiding unnecessary forced termination sessions.

When modelling this type of systems, we have a structure characterized

by two basic functional blocks: a main block which houses the set of servers, i.e., the system resources, plus a possible queue. On the other hand, there is a second block that houses the users who retry, usually called retrial orbit. Moreover, we consider the possibility that users are impatient and leave the system without being served. For the specific case of a monoservice cellular network, the system presents two orbits, one for new service requests and one for handovers, since the characteristics of these two types requests are different.

These systems can be characterized as a multidimensional continuous time Markov chain, CTMC. Dimensions represent the servers, or system resources, and the users in each of the retrial orbits of the system. In the case of studying systems with infinite population, we have systems in which the dimensions that represent the retrial orbits are infinite and moreover, they present state-dependent transitions in all dimensions. With these features it is impossible to solve the system in an exact way and it is necessary to use approximate models to obtain the state probabilities of the system.

Throughout this work we have developed approximate models in order to improve the performance of those models found in the specialized literature. We have developed a model, called *Finite Model*, FM, that belongs to the *Truncated models* category. This category is based on replacing the initial infinite state space for a finite one. We have also developed the *Model of state space Limitation*, LM, and *Homogenisation Models*, HM1 and HM2, all of them are *Generalized truncated* models. In this case, we approach the initial infinity state space that can not be solved —it is impossible to calculate the state probabilities of the system—, for another infinite one, but with some characteristics that make possible to solve the system.

Specifically, the LM model considers that the rate of retrials is infinite for certain states; on the other hand, models HM1 and HM2 are based on the homogenization of the state space from a given level of associated Quasi Birth and Death Process, QBD. The fact of keeping the infinite state space improves the accuracy of these models, compared to that obtained with models that

use a finite state space. These models were compared, in a generic scenario, with the most popular models in the literature. Results show that FM gets better results in terms of accuracy than the other *Truncated models* compared. Obviously, *Generalized truncated models* get better results than *Truncated models*. We emphasize the use of the HM2 model, which gets a very good trade off between accuracy and computational cost.

All these models are based on the calculation of the probabilities of state. Recently, however, an alternative approach to evaluate Markov processes, including those with an infinite state space, has appeared. It is called Extrapolation Value (VE). The main feature of this approach is that it considers the system as a Markov Decision Process, MDP. This solution has been adapted to retrial systems, obtaining an approximate model that is very versatile and it presents a very good performance, both in terms of accuracy and computational cost.

You may find other retrial systems that characterize a number of new applications such as VoIP or video conferencing services. These applications, in case of blocking, allow the user to retry the access with a lower number of resources requested. Thus, adaptive rate techniques have been developed to offer a greater or lesser quality according to the degree of congestion. Apart from these applications, we find those applications related to the transfer of electronic documents and that can be modelled as elastic traffic. In this work we have developed different mechanisms, working together with the admission control policy, to improve the efficiency of the network while ensuring a certain quality of service to the users of these applications. Specifically, a reserve policy has been developed, which gets a soft degradation in the performance of the rate adaptive flows when there is congestion in the system. Additionally, we can include a elastic traffic flow as best effort in order to utilize the resources that real-time users leave free, without affecting the quality of service obtained by the real time flow.

Resum

En l'última dècada les xarxes mòbils han experimentat un enorme creixement. Així mateix l'augment del nombre de serveis i l'ample de banda que requereixen fa necessari l'ús d'un correcte modelat i gestió de recursos. Aquest treball pretén ser una contribució al desenvolupament de models per a l'estudi i l'avaluació de la gestió de recursos radio en xarxes mòbils cel·lulars. Més concretament, s'ha pretès aprofundir en l'estudi de models de reintents. Estos models son de gran utilitat per a la caracterització de diferents aspectes de funcionament, com pot ser el modelat del comportament humà o la caracterització dels nous servicis oferts per les xarxes de comunicacions. Tradicionalment, s'ha entès per sistema de reintents aquell sistema en que els usuaris que son bloquejats, tracten d'accedir novament, després d'un temps d'espera. Esta és una característica pròpia del comportament humà que no s'ha d'obviar en el modelat de sistemes de comunicacions, ja que pot tenir un gran impacte en las prestacions —probabilitat de bloqueig, probabilitat de acabament forçós, etc.— ofertes pel sistema. Addicionalment, en les xarxes mòbils cel·lulars, per la seua estructura i característiques pròpies, es pot trobar aquest efecte també en els *handovers*. Així, d'acord amb l'estàndard de GSM, per exemple, mentre el mòbil es troba en l'àrea de *handover* —àrea de solapament entre la cobertura de dos o mes cèl·lules—, i sense que l'usuari s'adone, pot demanar recursos a la cèl·lula destí del *handover* un nombre limitat de voltes. Una correcta caracterització d'aquest procés millorarà les prestacions de la xarxa, evitant talls innecessaris de sessions en curs.

A l'hora de modelar aquest tipus de sistemes apareix una estructura ca-

racteritzada per dos blocs funcionals bàsics: un bloc principal que inclou el conjunt de servidors, es a dir, els recursos del sistema, més una possible cua d'espera. Per un altra banda, apareix el bloc on s'allotgen els usuaris que reintenten denominat, generalment, òrbita de reintents. A més a més, s'observa la possibilitat que els usuaris s'impacienten i abandonen el sistema sense haver estat servits. Per al cas concret d'una xarxa cel·lular monoservici tindrem dues òrbites, una per a peticions de servei noves i un altra per a *handovers*, ja que les característiques d'aquests dos tipus de peticions són diferents.

Aquests sistemes es poden caracteritzar com una cadena de Markov contínua en el temps, CTMC, multidimensional. On aquestes dimensions representaran, els servidors, o recursos de la cèl·lula, i els usuaris en cadascuna de les òrbites de reintents del sistema. En el cas d'estudiar sistemes de població infinita, ens trobem amb sistemes en que les dimensions que representen a les òrbites de reintents són infinites, i a més a més, amb transicions depenents de l'estat en totes les dimensions. Amb aquestes característiques és impossible resoldre el sistema de forma exacta i és necessari recórrer a models aproximats per a l'obtenció de les probabilitats d'estat del sistema

Al llarg d'aquest treball s'han desenvolupat diferents models aproximats amb la finalitat de millorar les prestacions d'aquells que podem trobar en la literatura especialitzada. S'ha desenvolupat un model al que hem denominat *Finite Model*, FM, pertanyent a la categoria dels *Truncated models*, basats a reemplaçar l'espai d'estats infinit inicial, per un altre que siga finit. També s'ha desenvolupat el *Model de Limitació* de l'espai d'estats, LM, i els *Models d'Homogeneïtzació*, HM1 i HM2 , pertanyents, tots ells, a la categoria dels *Generalized truncated models*. En aquest cas, s'aproxima un sistema infinit i que no es pot resoldre —en el sentit que resulta impossible calcular les probabilitats d'estat—, per un altre també infinit però que per les seues característiques sí es pot resoldre. En concret el model LM està basat a considerar que la taxa de reintents és infinita per a determinats estats; mentre que els models HM1 i HM2 estan basats en l'homogeneïtzació de l'espai d'estats a partir d'un determinat nivell del *Quasi Birth and Death Process*, QBD, associat. El fet de mantenir l'espai d'estats infinit permet millorar la precisió d'aquests

models enfront de la qual obtindríem amb els models que utilitzen un espai d'estats finit. Aquests models s'han comparat, en un escenari genèric, amb els models més coneguts de la literatura. Els resultats mostren com FM obté millors resultats en termes de precisió que la resta dels models de la categoria *Truncated models* comparats. Òbviament els models *Generalized truncated models* aconseguiran millors resultats que els *Truncated models*, i entre ells destaca el model HM2 que aconsegueix una molt bona relació precisió enfront de cost computacional. Tots aquests models estan basats en el càlcul de les probabilitats d'estat.

Recentment, no obstant això, ha aparegut una aproximació, denominada *Value Extrapolation* (VE), alternativa per a avaluar processos de Markov, inclosos aquells amb un espai d'estats infinit. La principal característica d'aquesta aproximació és que considera el sistema com un *Markov Decision Process*, MDP. S'ha adaptat aquesta solució a sistemes de reintents, obtenint-se un model aproximat molt versàtil i amb molt bones prestacions tant en termes de precisió com de cost computacional.

És possible trobar altre tipus de sistemes de reintents que caracteritzen tot un seguit de noves aplicacions com VoIP o serveis de videoconferència. Es tracta d'aplicacions que, en cas de bloqueig, permeten reintentar l'accés disminuint el nombre de recursos sol·licitat. Així, apareixen tècniques de *rate adaptive* on, segons el grau de congestió, s'ofereix un servei de major o menor qualitat. Per altra banda, apareixen les aplicacions relacionades amb la transferència de documents electrònics, que poden ser modelades com tràfic elàstic. En aquest treball s'han desenvolupat diferents mecanismes que, treballant juntament amb la política de control d'admissió, permeten millorar l'eficiència de la xarxa alhora que asseguren una determinada qualitat de servei als usuaris d'aquestes aplicacions. En concret, s'ha desenvolupat una política de reserva de recursos que aconsegueix una degradació suau de les prestacions dels diferents fluxos *rate adaptive* quan existeix congestió en el sistema. Addicionalment, s'ha vist com es pot incloure un flux de tràfic elàstic com *best-effort* amb la finalitat d'aprofitar els recursos que els fluxos de temps real deixen lliures sense que açò afecte a la qualitat de servei obtinguda pels

fluxos de temps real.

Resumen

En la última década las redes móviles han experimentado un enorme crecimiento. Asimismo el aumento del número de servicios y el ancho de banda que requieren hace necesario un correcto modelado y gestión de recursos. Este trabajo pretende ser una contribución al desarrollo de modelos para el estudio y la evaluación de la gestión de recursos radio en redes móviles celulares. Más concretamente, se ha pretendido profundizar en el estudio de modelos de reintentos. Estos modelos son de gran utilidad para la caracterización de diferentes aspectos de funcionamiento, como puede ser el modelado del comportamiento humano o la caracterización de los nuevos servicios ofrecidos por la redes de comunicaciones. Tradicionalmente, se ha entendido por sistema de reintentos aquel sistema en que los usuarios que son bloqueados, tratan de acceder de nuevo, tras un tiempo de espera. Esta es una característica propia del comportamiento humano que no debe obviarse en el modelado de sistemas de comunicaciones, puesto que puede tener un gran impacto en las prestaciones —probabilidad de bloqueo, probabilidad de terminación forzosa, etc.— ofrecidas por el sistema. Adicionalmente, en las redes móviles celulares, por su estructura y características propias, podemos encontrar este efecto también en los *handovers*. Así, de acuerdo con el estándar de GSM, por ejemplo, mientras el móvil se encuentra en el área de *handover* —área de solape entre la cobertura de dos o más células—, y sin que el usuario lo perciba, puede pedir recursos a la célula destino del *handover* un número limitado de veces. Una correcta caracterización de este proceso de reintento mejorará las prestaciones de la red, evitando cortes innecesarios de

sesiones en curso.

A la hora de modelar este tipo de sistemas nos aparece una estructura caracterizada por dos bloques funcionales básicos: un bloque principal que alberga el conjunto de servidores, es decir los recursos del sistema, más una posible cola de espera. Por otro lado aparece el bloque donde se alojan los usuarios que reintentan, denominado generalmente órbita de reintentos. Además se observa la posibilidad de que los usuarios se impacienten y abandonen el sistema sin haber sido servidos. Para el caso concreto de una red celular monoservicio tendremos dos órbitas, una para peticiones de servicio nuevas y otra para *handovers*, puesto que las características de estos dos tipos de peticiones son diferentes.

Estos sistemas se pueden caracterizar como una cadena de Markov continua en el tiempo, CTMC, multidimensional. Donde estas dimensiones representarán, los servidores, o recursos de la célula, y los usuarios en cada una de las órbitas de reintentos del sistema. En el caso de estudiar sistemas de población infinita, nos encontramos con sistemas en que las dimensiones que representan a las órbitas de reintentos son infinitas, y además, con transiciones dependientes del estado en todas las dimensiones. Con estas características es imposible resolver el sistema de forma exacta y es necesario recurrir a modelos aproximados para la obtención de las probabilidades de estado del sistema.

A lo largo de este trabajo se han desarrollado diferentes modelos aproximados con el fin de mejorar las prestaciones de aquellos que podemos encontrar en la literatura especializada. Se ha desarrollado un modelo al que hemos denominado *Finite Model*, FM, perteneciente a la categoría de los *Truncated models*, basados en reemplazar el espacio de estados infinito inicial, por otro que sea finito. También se ha desarrollado el Modelo de Limitación del espacio de estados, LM, y los Modelos de Homogeneización, HM1 y HM2, pertenecientes, todos ellos, a la categoría de los *Generalized truncated models*. En este caso, se aproxima un sistema infinito y que no se puede resolver -en el sentido de que resulta imposible calcular las probabilidades de estado-,

por otro también infinito pero que por sus características sí se puede resolver. En concreto el modelo LM está basado en considerar que la tasa de reintentos es infinita para determinados estados; mientras que los modelos HM1 y HM2 están basados en la homogeneización del espacio de estados a partir de un determinado nivel del *Quasi Birth and Death Process*, QBD, asociado. El hecho de mantener el espacio de estados infinito permite mejorar la precisión de estos modelos frente a la que obtendríamos con los modelos que utilizan un espacio de estados finito. Estos modelos se han comparado, en un escenario genérico, con los modelos más conocidos de la literatura. Los resultados muestran como FM obtiene mejores resultados en términos de precisión que el resto de los modelos de la categoría *Truncated models* comparados. Obviamente los modelos *Generalized truncated models* conseguirán mejores resultados que los *Truncated models*, y entre ellos destaca el modelo HM2 que consigue una muy buena relación precisión frente a coste computacional.

Todos estos modelos están basados en el cálculo de las probabilidades de estado. Recientemente, sin embargo, ha aparecido una aproximación, denominada *Value Extrapolation* (VE), alternativa para evaluar procesos de Markov, incluidos aquellos con un espacio de estados infinito. La principal característica de esta aproximación es que considera el sistema como un *Markov Decision Process*, MDP. Se ha adaptado esta solución a sistemas de reintentos, obteniéndose un modelo aproximado muy versátil y con muy buenas prestaciones tanto en términos de precisión como de coste computacional.

Es posible encontrar otro tipo de sistemas de reintentos que caracterizan toda una serie de nuevas aplicaciones como VoIP o servicios de videoconferencia. Se trata de aplicaciones que, en caso de bloqueo, permiten reintentar el acceso disminuyendo el número de recursos solicitado. Así, aparecen técnicas de *rate adaptive* en que, según el grado de congestión, se ofrece un servicio de mayor o menor calidad. Por otra parte, aparecen las aplicaciones relacionadas con la transferencia de documentos electrónicos, que pueden ser modeladas como tráfico elástico. En este trabajo se han desarrollado diferentes mecanismos que, trabajando junto con la política de control de admisión, permiten mejorar la eficiencia de la red a la vez que aseguran una determi-

nada calidad de servicio a los usuarios de estas aplicaciones. En concreto, se ha desarrollado una política de reserva de recursos que consigue una degradación suave de las prestaciones de los diferentes flujos *rate adaptive* cuando existe congestión en el sistema. Adicionalmente, se ha visto como se puede incluir un flujo de tráfico elástico como *best-effort* con el fin de aprovechar los recursos que los flujos de tiempo real dejan libres sin que esto afecte a la calidad de servicio obtenida por los flujos de tiempo real.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

A diferencia de lo que ocurre en las redes fijas, donde la enorme capacidad de transmisión disponible relega a un segundo plano la gestión eficiente de esta capacidad, en las redes móviles celulares nos encontramos con un medio de transmisión radio, cuya capacidad proviene del espectro radioeléctrico y éste, por sus características, es un bien escaso.

Estas restricciones han tratado de solventarse durante los últimos años, pudiéndose encontrar una gran cantidad de avances tecnológicos en este campo. Una de las primeras soluciones planteadas contemplaba la ampliación de la banda del espectro en el cual operan este tipo de redes. Sin embargo, esta solución no parecía la más adecuada ya que no solucionaba el problema sino que sólo lo retrasaba [RWO95]. Otras soluciones que han venido empleándose en los últimos años han sido la mejora en la eficiencia espectral mediante el uso de modulaciones más eficientes [FNO99], o la mejora del rendimiento mediante el uso de mecanismos de reutilización del espectro basados en la sectorización o el empleo de células cada vez más pequeñas [Lee91]. No obstante, el enorme crecimiento en el número de usuarios, unido a los cambios en los servicios ofrecidos por estas redes, que han pasado de aplicaciones de voz, a servicios de datos de gran exigencia en cuanto a capacidad, han hecho que la gestión eficiente recursos siga siendo

un asunto de gran importancia.

Este trabajo pretende ser una contribución al desarrollo de modelos para el estudio y la evaluación de la gestión de recursos radio en redes móviles celulares. Más concretamente, se ha profundizado en el estudio de sistemas de reintentos. Estos, en sus diferentes modalidades, resultan de interés pues permiten caracterizar diferentes aspectos del funcionamiento de las redes móviles celulares. Una correcta caracterización de la red nos permitirá, a su vez, diseñar redes de comunicaciones capaces de ofrecer cierta calidad de servicio de forma eficiente.

Tradicionalmente, se ha entendido por sistema de reintentos a aquel en que los usuarios que han sido bloqueados por el sistema tratan de acceder al mismo tras un tiempo de espera. Este tipo de comportamiento se puede encontrar en cualquier tipo de red de comunicaciones, y las redes móviles celulares no van a ser una excepción. Además, se ha de tener en cuenta que, debido al número creciente de usuarios, así como a la complejidad de estas redes, el comportamiento de los usuarios en general, y el fenómeno de los reintentos en particular, tiene un impacto en las prestaciones de la red que no podemos obviar [TGM97]. En el caso de las redes móviles celulares, por su estructura y características propias, se puede encontrar este efecto no sólo debido al comportamiento humano, sino que también puede deberse a la gestión de los *handovers*[1] [ODEa02]. Cuando un usuario realiza un *handover* es necesario que la célula destino disponga de suficientes recursos libres para poder cursar dicha petición, si no es así, la propia red y, más en concreto, el terminal, reintentará el acceso repetidamente mientras se encuentre en el área de *handover*. Este tipo de reintentos está incluido en el estándar de GSM (*Global System for Mobile Communications*) [MP92], en que se especifica un número máximo de reintentos automáticos consecutivos [ODEa02], sin que el usuario intervenga ni se percate de lo que está ocurriendo.

[1] A pesar de que la palabra *handover* se traduce a la lengua española como ”traspaso” y el diccionario de la lengua española [Esp03] lo define como el ”traslado de algo de un lugar a otro", se ha preferido mantener el término anglosajón, por estar más extendido entre la comunidad científica hispanohablante.

La mayor parte del trabajo recogido en esta Tesis, capítulos 2 al 8, hacen referencia a este tipo de sistemas de reintentos. Pero además, se ha decidido incorporar, aunque sea de forma resumida, el estudio de otro tipo de sistemas de reintentos aplicado a la caracterización de aplicaciones. Se ha tomado esta decisión por ser un campo en el que se ha trabajo bastante durante estos últimos años y que ha venido avalado por los resultados obtenidos plasmados en diferentes publicaciones, pero también por ser una línea de investigación que puede ser muy interesante para el diseño de las nuevas redes de comunicaciones. Esta parte del trabajo parte de los conceptos desarrollados por Kaufman y Roberts [Kau92b] a principios de los años 80 para el análisis de sistemas multitasa. Estos trabajos han servido de base para el estudio de toda una serie de nuevas aplicaciones que han experimentado una alta penetración en las redes de comunicaciones durante los últimos años. Se trata de aplicaciones que, en caso de bloqueo, permiten reintentar el acceso disminuyendo el número de recursos solicitado. Dentro de este tipo de aplicaciones podemos encontrar dos posibilidades. Las aplicaciones *rate adaptive* [CPOG04], en que se establece una serie de valores, en general discretos, para definir el número de recursos que solicita una sesión. Según el grado de congestión, se utilizará un valor mayor o menor de recursos. Así, puesto que la duración de la sesión no cambia, se ofrece un servicio de mayor o menor calidad. Este tipo de modelo resulta de especial interés para tratar aplicaciones como VoIP o servicios de videoconferencia para las que se han desarrollado codecs de audio y video que permiten adaptar la tasa de servicio a las condiciones de la red [Cas01]. Por otra parte, aparecen las aplicaciones relacionadas con la transferencia de documentos electrónicos, que pueden ser fácilmente modeladas como tráfico elástico. En este tipo de tráfico la modificación de la tasa de transmisión va unida a la variación del tiempo de duración de la sesión de forma que la cantidad total de información a transmitir se mantenga constante. Un trato correcto de este tipo de aplicaciones puede ser de vital importancia para hacer un uso eficiente de los recursos. Esto es especialmente importante en el caso de redes móviles celulares, donde los recursos son escasos y existe una alta variabilidad en la disponibilidad de los mismos.

Estructura de la tesis

En el capítulo 2 se repasan brevemente los fundamentos de los sistemas de reintentos, prestando especial atención a su modelado y las distintas formas de afrontar su resolución. Los siguientes capítulos están dedicados al estudio de los modelos aproximados, que se presentan como el principal y más utilizado método de resolución de sistemas de reintentos. Dentro de los modelos aproximados podemos clasificar las diferentes soluciones en Aproximaciones, Modelos Truncados y Modelos Truncados Generalizados. De entre los tres tipos de modelos aproximados existentes, en este trabajo nos hemos centrado en los dos últimos tipos. Así, mientras que en el capítulo 3 se presentan las principales aproximaciones existentes en la literatura, en los capítulos 4 y 5, a parte de resumir la bibliografía existente en el ámbito de los modelos truncados y los truncados generalizados, respectivamente, se han desarrollado modelos propios con el fin de mejorar la eficiencia en la resolución de sistemas de reintentos. En el capítulo 6 podemos encontrar un análisis de prestaciones de los diferentes modelos desarrollados frente a las soluciones existentes. Asimismo, en el capítulo 7 se ha introducido el estudio de un modelo novedoso en cuanto que no se basa en el cálculo de las probabilidades de estado, sino que plantea el sistema como un *Markov Decision Process* (MDP).

En el capítulo 8 se aplican los modelos desarrollados a un escenario de redes móviles celulares. Mientras que en los capítulos anteriores todo el estudio llevado a cabo se ha realizado sobre un escenario genérico, en este capítulo se introducen los reintentos en un escenario propio de una red celular. De este modo, el escenario estudiado introduce tanto sesiones nuevas como de *handover* —con sus correspondientes reintentos— y tiene en cuenta el uso de mecanismos de control de admisión que permitan la priorización de los *handovers* sobre las sesiones nuevas. Este capítulo terminará con un estudio del efecto de los reintentos en un escenario celular. Este estudio muestra como las suposiciones realizadas comúnmente respecto a los reintentos a la hora de modelar redes de comunicaciones no resultan muy satisfactorias; motivando, de este modo, la necesidad de utilizar sistemas de reintentos a tal

efecto. Nótese que los estudios realizados tanto en este capítulo como en los capítulos 4–7 son totalmente analíticos. Aun siendo conscientes de que este tipo de modelos no puede capturar toda la complejidad de un sistema móvil celular, sí pueden resultar de interés, ya que permiten descubrir tendencias de funcionamiento, así como resultados cualitativos que pueden ayudar a la mejor comprensión del funcionamiento del sistema.

Por otra parte, el capítulo 9 introduce un breve estudio de la aplicación de sistemas de reintentos a la gestión de las nuevas aplicaciones. Este apartado parte de los estudios de escenarios multitasa para desarrollar sistemas de gestión que, trabajando junto con mecanismo de control de admisión [GB06] como *Multiple Guard Channel* (MGC), permitan una gestión eficiente en entornos con aplicaciones de tráfico *rate adaptive* y elástico.

Por último, el capítulo 10 presenta un resumen del trabajo realizado, destacando sus principales conclusiones. Asimismo se sugieren de manera genérica posibles líneas de trabajo futuras.

Capítulo 2

Sistemas de reintentos, introducción a su modelado y resolución

Los sistemas de reintentos se presentan como una interesante alternativa complementaria a los clásicos modelos de pérdidas que se suelen utilizar a la hora de modelar redes de comunicaciones. Generalmente, cuando se pretende diseñar o evaluar una de estas redes, es habitual considerar que cualquier usuario que llegue al sistema y encuentre todos los recursos ocupados, abandonará el sistema o se unirá a una cola de espera. Sin embargo, esta consideración no tiene en cuenta el hecho de que un usuario, tras ser bloqueado, no abandone el sistema definitivamente, sino que abandone temporalmente el área de servicio para reintentar el acceso tras un tiempo aleatorio. Este tipo de comportamiento, conocido como reintento, puede encontrarse en sistemas tan conocidos como las redes de telefonía fija tradicional, pero también en las redes móviles celulares, entre las que cabe destacar las redes de acceso móvil como GSM (*Global System for Mobile Communications*) [MP92], GPRS (*General Packet Radio Service*)[BVE99] y UMTS (*Universal Mobile Telecommunications System*) [MK00]. En otras tecnologías de reciente aparición como IEEE 802.16 [Sta07], más conocido como WiMAX y que recientemente ha incorporado movilidad en su estandard [Sta07], así como IEEE 802.20 [BXG07]

(*Mobile Broadband Wireless Access*) también es posible encontrar este tipo de comportamientos.

Debe tenerse en cuenta además, que obviar la existencia de los reintentos a la hora de modelar el sistema puede tener un efecto negativo en las prestaciones del sistema. De este modo en [TGM97] se demuestra, para el caso de redes móviles celulares, que no tengan en cuenta los reintentos puede producir, en situaciones de sobrecarga, un efecto de bola de nieve en los procesos de llegada de peticiones al sistema que producirá una fuerte degradación de la calidad de servicio experimentada por los usuarios. Esta degradación se produce porque la red no espera la aparición de reintentos y por lo tanto, se ve sorprendida por una carga adicional que no se había tenido en cuenta durante la fase de diseño. Otra posibilidad a la hora de caracterizar los reintentos consiste en considerarlos como parte del flujo usuarios primarios. De esta forma, el sistema tiene en cuenta, desde el primer momento, la existencia de una carga adicional producida por los reintentos. Sin embargo, esta caracterización tampoco resulta adecuada puesto que, al tratar como peticiones primarias a reintentos, se puede llegar a contar más pérdidas de las que existen en realidad. Por ejemplo, si tomamos un usuario que tras reintentar x veces abandona el sistema sin obtener servicio, esta solución consideraría que se han perdido $x+1$ peticiones, cuando en realidad sólo se ha perdido una. Es por este hecho que este tipo de soluciones conlleva un sobredimensionamiento de los sistemas. En el caso de las redes inalámbricas, esto puede llegar a ser muy perjudicial al tratarse de redes con una gran escasez de recursos.

Se puede concluir que los reintentos van a tener un efecto no despreciable en el sistema, y que pueden llegar a tener una considerable influencia negativa en los parámetros de prestaciones del sistema. De ahí la necesidad de introducir este tipo de comportamiento en el modelado de redes de comunicaciones, y para ello se recurrirá al uso de colas con reintentos [Coh57].

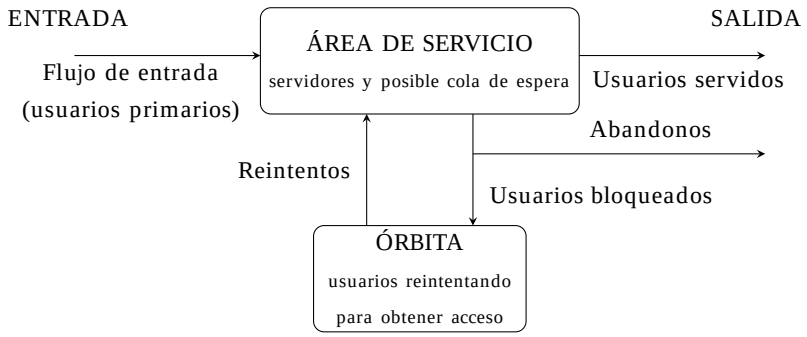


Figura 2.1: Estructura general de una cola de reintentos.

2.1 Estructura de un sistema de reintentos

La estructura básica de una cola de reintentos se puede observar en la figura 2.1. Así aparecen dos bloques funcionales básicos: un bloque principal que alberga el conjunto de servidores, es decir los recursos del sistema, más una posible cola de espera. Por otro lado aparece el bloque donde se alojan los usuarios que reintentan, denominado generalmente órbita de reintentos.

Algunas veces a estos dos bloques básicos se les añaden características extra, como la posibilidad de que los usuarios se impacienten y abandonen el sistema sin haber sido servidos. Esta propiedad resulta muy interesante para nuestro estudio y es por ello que se ha decidido incorporarla desde un principio.

Aunque las colas de reintentos se han venido modelando a partir de dos estructuras básicas como son la $M/M/C$[1] con reintentos [Fal90, YT87] y la $M/G/1$ [FT97, CCL95], para el caso que nos ocupa nos centraremos únicamente en las estructuras $M/M/C$ puesto que nos permiten el estudio de

[1]Se ha seguido la notación de Kendall [Tij03], según la cual una cola puede describirse mediante la notación $A/B/C/K/N/D$, donde A = distribución del proceso de llegadas, B = distribución del tiempo de servicio, C = número de servidores, K = capacidad el sistema (servidores + cola de espera), N = tamaño de la población y D = disciplina de la cola

sistemas multiservidor como los que se van a encontrar en las redes de comunicaciones. Las colas $M/G/C$ resultan más complejas y, por otra parte, suponer un tiempo de servicio exponencial no conlleva una perdida importante de precisión en la modelización del sistema. Así en las colas $M/G/C/C$ la probabilidad de pérdidas es inmune a la distribución del tiempo de servicio, mientras que en las colas $M/G/C/\infty$ aunque la probabilidad de demora sí depende de la distribución de servicio, el error que se comete es despreciable puesto que esta probabilidad suele tomar un valor pequeño. En [AGC08, Art99a, Art99b, GC06] se puede encontrar una revisión de los principales trabajos aparecidos en la literatura.

El modelo básico de una cola $M/M/C$ con reintentos es el siguiente: Se considera un sistema multiservidor en el cual los usuarios primarios (peticiones nuevas) llegan según un proceso de Poisson, de tasa λ, a un sistema que dispone de C servidores idénticos y cuyo tiempo de servicio está distribuido según una ley exponencial de tasa μ. Si un usuario primario llega al sistema y encuentra algún servidor libre, lo ocupará automáticamente, abandonando el sistema tras el tiempo de servicio. Por el contrario, si no encuentra ningún servidor disponible, el usuario se unirá a la órbita de reintentos. Los usuarios en la órbita de reintentos intentarán el acceso a los servidores tras un tiempo exponencial de tasa μ_r. Si un reintento encuentra algún servidor libre, lo ocupará inmediatamente. Por el contrario, en caso de encontrar todos los servidores ocupados, el usuario podrá abandonar el sistema con una determinada probabilidad, P_i, o volver a la órbita de reintentos con la probabilidad complementaria, $1 - P_i$.Con este modelo se asegura la existencia de por lo menos un reintento.

Obsérvese la diferencia en el comportamiento de las colas de reintentos frente a los típicos sistemas con colas de espera como el que podemos encontrar en [Bar04, HR86]. En este último caso, los servidores permanecen en activo hasta que la cola se vacía. Por el contrario, en el caso de sistemas con reintentos, como el descrito en la figura 2.2, cuando un servidor termina un servicio, permanecerá inactivo un determinado tiempo aleatorio y no volverá a entrar en servicio, bien hasta que alguno de los usuarios en la órbita rein-

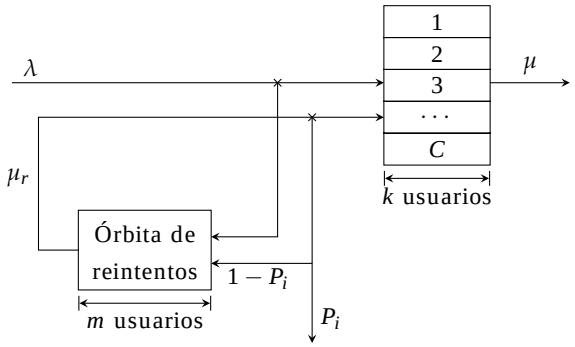


Figura 2.2: Modelo básico de sistema de reintentos.

tente, bien hasta que llegue al sistema una petición de un usuario primario. Así, por ejemplo, para el caso de que $C = 1$, el tiempo transcurrido desde que el servido queda inactivo hasta que empieza a servir a un nuevo usuario será una variable aleatoria exponencialmente distribuida y de tasa $\lambda + j\mu_r$, donde j indica el número de usuarios en la órbita de reintentos. Se asume que los tiempos de llegada, servicio y reintento son mutuamente independientes.

El sistema de la figura 2.2 se puede representar como una cadena de Markov continua en el tiempo (CTMC: *Continuous Time Markov Chain*) bidimensional (k, m), donde k es el número de servidores ocupados y m representa el número de usuarios en la órbita de reintentos. La figura 2.3 muestra el espacio de estados de esta cadena, determinado por el semiespacio $\{0, \ldots, C\} \times \mathbb{Z}_+$. Ordenamos los estados como

$$S = \{(0,0), \ldots, (C,0), (0,1), \ldots, (C,1), \ldots\},$$

el generador infinitesimal de esta cadena tiene una estructura tridiagonal a bloques, típica de los procesos *QBD* (*Quasy Birth-and-Death*) [Neu81]:

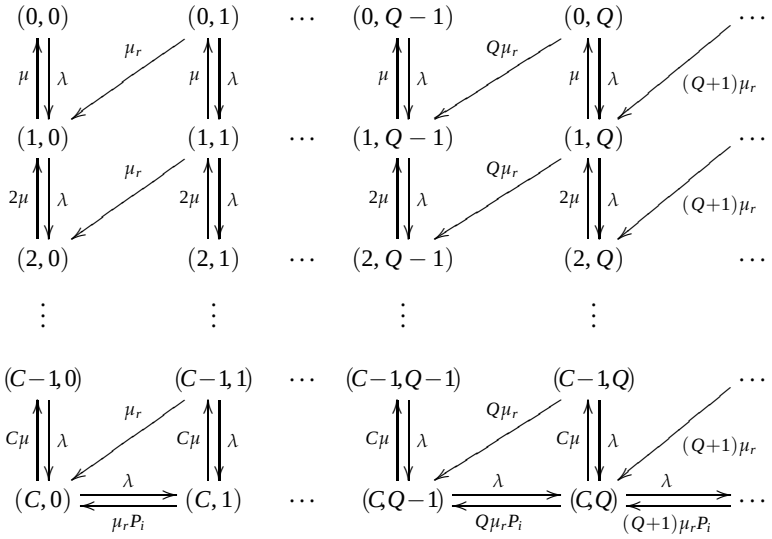


Figura 2.3: Diagrama de transiciones del modelo básico.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \mathbf{A}_0^{(1)} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{A}_2^{(2)} & \mathbf{A}_1^{(2)} & \mathbf{A}_0^{(2)} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \dots \end{bmatrix},$$

donde las matrices $\mathbf{A}_0^{(m)}$, $\mathbf{A}_1^{(m)}$ y $\mathbf{A}_2^{(m)}$, de tamaño $(C+1) \times (C+1)$, vienen

dadas por:

$$\mathbf{A_0^{(m)}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \lambda \end{bmatrix},$$

$$\mathbf{A_1^{(m)}} = \begin{bmatrix} * & \lambda & 0 & \dots & 0 \\ \mu & * & \lambda & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2^{(m)}} = \begin{bmatrix} 0 & m\mu_r & 0 & \dots & 0 \\ 0 & 0 & m\mu_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & m\mu_r \\ 0 & 0 & 0 & \dots & m\mu_r P_i \end{bmatrix}.$$

Los asteriscos que aparecen en la diagonal principal de $\mathbf{A_1^{(m)}}$ son valores que hacen que la suma de los elementos de la fila correspondiente del generador infinitesimal $\mathbf{Q}$ sea cero.

El objetivo es calcular el vector de probabilidades de estado en régimen permanente, $\boldsymbol{\pi}$, a partir del cual se podrán calcular diferentes parámetros de prestaciones del sistema. Para obtener las probabilidades de estado se resolverán las ecuaciones $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, junto con la condición de normalización, $\boldsymbol{\pi}\mathbf{e} = 1$, donde $\mathbf{e}$ es un vector columna en el que todos los elementos son iguales a la unidad. Denominando $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots)$ al vector fila de probabilidades de estado en régimen permanente donde $\boldsymbol{\pi}_m = (\pi(0,m), \dots, \pi(C,m))$, podemos escribir dichas ecuaciones como:

$$\begin{aligned} \boldsymbol{\pi}_0 \boldsymbol{A}_1^{(0)} + \boldsymbol{\pi}_1 \boldsymbol{A}_2^{(1)} = 0, \quad m = 0 \\ \boldsymbol{\pi}_{m-1} \boldsymbol{A}_0^{(m-1)} + \boldsymbol{\pi}_m \boldsymbol{A}_1^{(m)} + \boldsymbol{\pi}_{m+1} \boldsymbol{A}_2^{(m+1)} = 0, \quad m \geq 1 \end{aligned} \tag{2.1}$$

Obtenidas las probabilidades de estado, se calcularán los diferentes parámetros de mérito. Uno de los parámetros más comunes es la probabilidad de bloqueo, P_b, definida como la probabilidad de tener todos los servidores ocupados[2]. Además existen otros parámetros de mérito de interés:

- Probabilidad de servicio inmediato (P_{si}): es la probabilidad de que un usuario acceda a los servidores del sistema en su primer intento. Es decir, el porcentaje de usuarios que son atendidos de forma inmediata.
- Probabilidad de servicio demorado (P_{sd}): es la probabilidad de que un usuario obtenga servicio, pero no en su primer intento. Es decir, es el porcentaje de usuarios que son atendidos tras un primer intento fallido.
- Probabilidad de no servicio (P_{ns}): es la probabilidad de abandonar el sistema sin haber obtenido servicio. De forma análoga a las anteriores probabilidades, es el porcentaje de usuarios que no son atendidos. Nótese que $P_{ns} = 0$ cuando no se tiene en cuenta el fenómeno de la impaciencia ($P_i = 0$).

Obviamente debe cumplirse que $P_{si} + P_{sd} + P_{ns} = 1$. Estos parámetros pueden expresarse en función de una serie de tasas. Entre ellas tenemos la tasa ofrecida al sistema, R_o, la tasa de primeros intentos con éxito, $R_{1,s}$, y la tasa de primeros intentos fallidos, $R_{1,f}$; cumpliéndose que $R_o = R_{1,s} + R_{1,f}$. La tasa de primeros intentos fallidos puede, a su vez, descomponerse en la tasa de reintentos exitosos, $R_{r,s}$, y la tasa de abandono, R_{ab}, $R_{1,f} = R_{r,s} + R_{ab}$. La tasa de abandono, a su vez, puede definirse como el producto de la tasa

[2] Nótese que la P_b puede referirse a *Time Congestion*, *Call Congestion* o *Traffic Congestion* [Ive06]. En este caso la P_b se refiere al *Time Congestion*, es decir, al porcentaje de tiempo que todos los servidores se encuentran ocupados.

de reintentos fallidos, $R_{r,f}$ por la probabilidad de impaciencia, P_i, es decir, $R_{ab} = P_i R_{r,f}$. De forma similar a la tasa ofrecida al sistema, se puede definir la tasa de reintentos, R_r como la suma de la tasa de reintentos exitosos más la tasa de reintentos fallidos, $R_r = R_{r,s} + R_{r,f}$. A partir de estas tasas, que serán calculadas en los capítulos correspondientes, se pueden calcular las probabilidades descritas anteriormente como:

$$P_{si} = \frac{R_{1,s}}{R_o} \quad ; \quad P_{sd} = \frac{R_{r,s}}{R_o} \quad ; \quad P_{ns} = \frac{R_{ab}}{R_o} = \frac{P_i R_{r,f}}{R_o} \tag{2.2}$$

Otro parámetro que se suele utilizar es el número medio de usuarios en la órbita de reintentos, definido como $N_{ret} = E(m)$, donde $E(x)$ representa la esperanza de la variable x.

Todo el desarrollo realizado hasta el momento se corresponde al del modelo básico de las colas de reintento de tipo $M/M/C$ mostrado en la figura 2.2. Nótese, sin embargo, que siendo este tipo de colas de reintentos uno de los más utilizados, existen múltiples variaciones posibles sobre el mismo. Entre estas variaciones podemos encontrar por ejemplo los sistemas de población finita como los propuesto en [HL98, Jan97, TGM97]. Otra posibilidad se puede encontrar en [BDK89] donde se analiza la situación en que si la respuesta a una determinada petición no llega antes de un *time-out* especificado, el procesador reintenta la petición. También, aparecen los sistemas donde los reintentos son constantes, como los que utilizan Choi et al. en [Cho92] para el modelado de redes de comunicaciones que hacen uso del protocolo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) y que se caracterizan por el hecho de que tasa de salida de la órbita de reintentos es fija y no depende del número de usuarios en la misma. Cualquiera de estas variaciones se puede derivar del desarrollo presentado para el modelo básico presentado en esta sección.

2.2 Resolución de sistemas de reintentos

Los sistemas de reintento multiservidor con población infinita presentan dos características fundamentales: tener un espacio de estados infinito en una dimensión y la aparición de transiciones dependientes del estado tanto en la primera dimensión como en la segunda, tal y como se observa en la figura 2.3. Este tipo de estructuras, en que aparece heterogeneidad en las dos dimensiones se conocen como *level dependent QBD* [Gre01], hecho que complica sustancialmente la resolución del sistema. Para los modelos con $C \leq 2$ las probabilidades de estado satisfacen un conjunto de ecuaciones de nacimiento y muerte, de forma que es posible obtener una solución explicita en forma de funciones hypergeométricas generalizadas [FT97, Art96, GCR99]. Esto no ocurre para el caso de $C > 2$ y como consecuencia, las probabilidades de estado no se pueden expresar de forma tratable. Aunque existen algunos artículos que tratan de resolver analíticamente sistemas con reintentos con un número de servidores arbitrario, como son los casos de [Pea89] donde se hace uso de fracciones continuas extendidas, o de [Coh57] con integrales de contorno, estos resultados no son de fácil aplicabilidad. Por lo tanto, en el caso de sistemas de reintentos multiservicio con $C > 2$, es necesario recurrir a métodos numéricos [TB95, LR99] y/o modelos aproximados como pueden ser, entre otros, [Fal83, NR90, Ste99].

2.2.1 Modelos aproximados

Dentro de los modelos aproximados podemos categorizar las diferentes soluciones existentes en tres clases, por un lado las aproximaciones y por otro, los modelos truncados (*truncated models*) y los modelos truncados generalizados (*generalized truncated models*):

- Aproximaciones: Esta categoría incluye aquellas soluciones en las que el modelo original es sustituido por otro simplificado que permite obtener las probabilidades de estado en régimen permanente. Sin embargo,

dicha simplificación está basada en suposiciones de carácter local que hacen que la solución sólo sea válida para ciertos valores de los parámetros del sistema o para casos extremos, como puedan ser situaciones de sobrecarga, alta o baja tasa de reintentos, etc.

- *Truncated models*: Estos modelos se basan en reemplazar el espacio de estados infinito inicial, S, por otro, S', que sea finito.

- *Generalized truncated models*: En este caso, se aproxima un sistema infinito y que no se puede resolver —en el sentido de que resulta imposible calcular las probabilidades de estado—, por otro también infinito pero que por sus características sí se puede resolver. El hecho de mantener el sistema infinito permite mejorar la precisión de estos modelos frente a la que obtendríamos con los modelos truncados.

Indicar que la diferencia entre la categoría de Aproximaciones y las otras dos categorías no siempre resulta clara, puesto que todos los modelos son, en realidad, aproximaciones y sus resultados van a ser, generalmente, mejores en un dominio de funcionamiento que en otro.

2.2.2 Métodos matemáticos

Muchos de los modelos propuestos resultan ser procesos de Markov con un generador infinitesimal, $\mathbf{Q}$, con una estructura tridiagonal en la que los elementos de la matriz son a su vez matrices, es decir, tendremos un proceso *QBD* [Neu81]. Los procesos *QBD* se pueden definir como cadenas de Markov con un espacio de estados $(i,l)|1 \leq i \leq n, l > 0$, donde el espacio de estados se divide en niveles, y cada nivel l dispone de n fases. En este tipo de procesos sólo se permiten las transiciones entre niveles adyacentes o dentro de un mismo nivel, lo que produce generadores $\mathbf{Q}$ de forma tridiagonal. En el caso de que las submatrices del generador infinitesimal que definen los diferentes niveles sean iguales para todos los niveles, diremos que el proceso *QBD* es homogéneo o *level-independent*. Por contra, en el caso de que al cambiar de

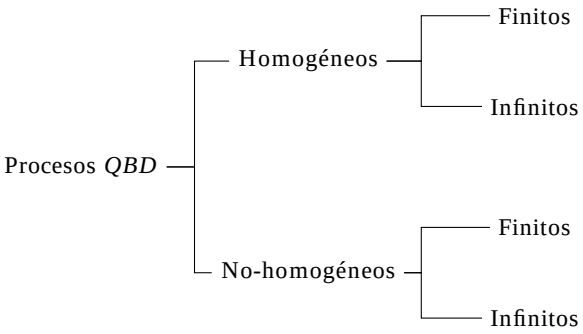


Figura 2.4: Tipos de procesos *QBD*.

nivel el contenido de las submatrices también cambie, nos encontraremos antes un *QBD* no homogéneo o *level-dependent*. Dentro de estos dos tipos de *QBD* podemos encontrar otra subdivisión, entre procesos con espacios de estados finitos e infinitos, lo que nos da la clasificación indicada en la figura 2.4.

En la literatura existe una gran cantidad de trabajos dedicados a la resolución algorítmica de este tipo de procesos. Estos algoritmos aprovechan las características especiales del generador infinitesimal del proceso *QBD* para obtener una resolución más eficiente. Los procesos *QBD* homogéneos han sido ampliamente tratados en la literatura, tanto para el caso finito como para el infinito. Para los procesos *QBD* homogéneos y finitos encontramos el algoritmo propuesto por Hajek en [Haj82]. Para el caso de procesos homogéneos e infinitos mencionar el trabajo de Neuts [Neu81] donde se desarrollan métodos para determinar la recurrencia positiva de la cadena Markov del proceso y calcular la distribución de equilibrio. Neuts también considera el caso de procesos *QBD* homogéneos en que aparece algún comportamiento límite no-homogéneo. Posteriormente Latouche y Ramaswami [LR93] desarrollaron otro algoritmo para calcular la distribución de equilibrio de este tipo de procesos con menor complejidad computacional.

Para el caso no-homogéneo, la mayoría de soluciones propuestas se cen-

tran en el caso finito. Así encontramos los algoritmos propuestos por Gaver, Jacobs y Latouche [GJL84] donde se desarrolla una solución generalizada para obtener la distribución de equilibrio de un *QBD* no-homogéneo con un espacio de estados finito. Otros métodos para resolver este tipo de proceso los encontramos en los trabajos de Ye y Li [YL94] donde se presenta el *Folding Algorithm* o en [Ser02] donde se presentan varios métodos de resolución que resultan muy eficientes. Para el caso de *QBDs* no-homogéneos e infinitos, únicamente encontramos el método propuesto por Bright y Taylor en [TB95] —se puede encontrar un resumen de este método en el apéndice C.1—. Este método realiza una truncación del espacio de estados a partir de un determinado nivel, e introduce el efecto de los niveles posteriores mediante el uso de una aproximación basada en el decrecimiento de las probabilidades de estado conforme aumenta el nivel.

Los siguientes capítulos están dedicados al estudio de diferentes modelos aproximados, estudiando su complejidad y coste de resolución. Este estudio servirá de base a la caracterización de los reintentos en redes móviles celulares que se abordará en el capítulo 8.

Capítulo 3

Aproximaciones

Posiblemente, la solución más intuitiva a la hora de resolver sistemas de reintentos, es considerar estos reintentos como parte del flujo entrante, es decir, como si se tratara de otro usuario nuevo / primario que trata de obtener acceso a los servidores. Esta aproximación, conocida como *Loss Model*, e introducida por Falin [FT97], permite tratar al sistema como un modelo de pérdidas del tipo $M/M/C/C$. Este modelo se puede resolver de forma inmediata haciendo uso de la distribución de Erlang's B [Kle75]:

$$\pi(i) = \frac{A^i/i!}{\sum_{k=0}^{C} A^k/k!} \qquad 0 \leq i \leq C, \tag{3.1}$$

donde $A = \Lambda/\mu$, con $\Lambda = \lambda + r$. Nótese que la tasa de llegadas, Λ, está compuesta por la tasa de usuarios primarios (λ) más la tasa global de reintentos, r, que se define como $r = N_{ret}\mu_r$, con N_{ret} el número medio de usuarios en la órbita de reintentos. Partiendo de la relación:

$$\lambda = \Lambda(1 - P_b) \quad \text{donde} \quad P_b = \pi(C) = \frac{A^c/c!}{\sum_{k=0}^{C} A^k/k!}, \tag{3.2}$$

Para poder calcular las probabilidades de estado en régimen permanente y así, los diferentes parámetros de prestaciones es necesario ajustar el valor de r. Para ello se hace uso de un proceso iterativo que consta de los siguientes pasos:

- Se da un valor a r, a partir del cual se obtiene el valor de tasa de llegadas como $\Lambda = \lambda + r$.

- Se calcula la probabilidad de bloqueo del sistema y, a partir de ésta un nuevo valor de λ' mediante las ecuaciones que se muestran en (3.2).

- Será necesario reajustar el valor de r hasta que $(\lambda - \lambda')/\lambda < \epsilon$.

Ajustada la tasa de reintentos, r, se pueden obtener las probabilidades de estado haciendo uso de la ecuación (3.1) y la probabilidad de bloqueo como $P_b^{(loss)} = \pi(C)$.

Este modelo ofrece una solución sencilla a un problema complejo. No obstante, aunque ofrece buenos resultados para una tasa de reintentos baja, cuando esta tasa crece se introduce un error que puede ser considerable. Este efecto se incrementa conforme aumenta el número de servidores, C.

Para valores elevados de la tasa de reintentos, una aproximación que genera mejores resultados que el *Loss Model* es el uso de un *Delay Model* de espera infinita: $M/M/C$. La probabilidad de que un usuario se encuentre todos los servidores ocupados y tenga que unirse a la cola de espera viene dada por la fórmula de Erlang C [Kle75]:

$$P_b^{(\infty)} = \pi(C) = \frac{\frac{(C\rho)^C}{C!}\frac{1}{1-\rho}}{\sum_{k=0}^{C-1}\frac{(C\rho)^k}{k!} + \frac{(C\rho)^C}{C!}\frac{1}{1-\rho}}$$

con $\rho = \lambda/C\mu < 1$.

Si el *Loss Model* obtiene resultados precisos para tasas de reintentos bajas, el uso del *Delay Model* permite obtener resultados adecuados cuando la tasa de reintento es alta. Habiendo observado esto, Falin [FT97] propone un modelo basado en la combinación de estos dos modelos. Este modelo, llamado *Interpolación*, calcula la probabilidad de bloqueo interpolada a partir de la probabilidad de bloqueo calculada con el *Loss Model*, $P_b^{(loss)}$, y la calculada

con el *Delay Model*, $P_b^{(\infty)}$, como:

$$P_b^{(Int)} \approx \frac{(C - C\rho)P_b^{(loss)} + \mu_r(1 - P_b^{(loss)})P_b^{(\infty)}}{(C - C\rho) + \mu_r(1 - P_b^{(loss)})}.$$

Los sistemas de Erlang B y Erlang C presentan características muy diferentes, sin embargo es posible interrelacionarlos a través de la probabilidad de no servicio inmediato, es decir, la probabilidad de encontrar todos los servidores ocupados, de forma similar a lo que se observa en [Hil79, Nes79]. Es esta probabilidad, a la que hemos denominado probabilidad de bloqueo, la que ha permitido combinar estos dos modelos en el de interpolación.

Otras soluciones se basan en la suposición de que *Returning Customers See Time Averages* (RTA). Es el caso de la aproximación propuesta por Greenberg y Wolff en [GW87] para la resolución de un escenario como el que se muestra en la figura 3.1, la cual muestra la posibilidad de abandono de los usuarios primarios (con probabilidad P_i^1), frente al modelo de la figura 2.3 donde se asumía $P_i^1 = 0$. Cuando $P_i^1 = P_i$ tenemos un sistema orbital geométrico. Además cuando $P_i = 0$ tenemos un sistema orbital infinito. En este caso el sistema será estable siempre que $\lambda(1 - P_i^1) < C\mu$. Resulta inmediato verificar que cuando $P_i > 0$ el sistema siempre es estable.

Sea $R(k)$ el porcentaje de usuarios que reintentan y se encuentran el sistema en estado k; M_R la tasa, en régimen permanente, con la que llegan los reintentos al sistema. Sea $\pi(k)$ el porcentaje de tiempo que el sistema de capacidad N tiene k usuarios en el mismo, con $0 \leq k \leq N$. Las ecuaciones de balance del flujo de probabilidades que entran y salen de un estado se pueden expresar como:

$$\lambda\pi(k) + M_R R(k) = (k+1)\mu\pi(k+1) \qquad 0 \leq k \leq N-1 \tag{3.3}$$

Para $k = N$ tenemos:

$$\lambda\pi(N)(1 - P_i^1) + M_R R(N)(1 - P_i) = M_R. \tag{3.4}$$

En la ecuación (3.3) el primer término refleja la tasa de sesiones primarias que llegan al sistema y el segundo término la tasa de reintentos ofrecida al

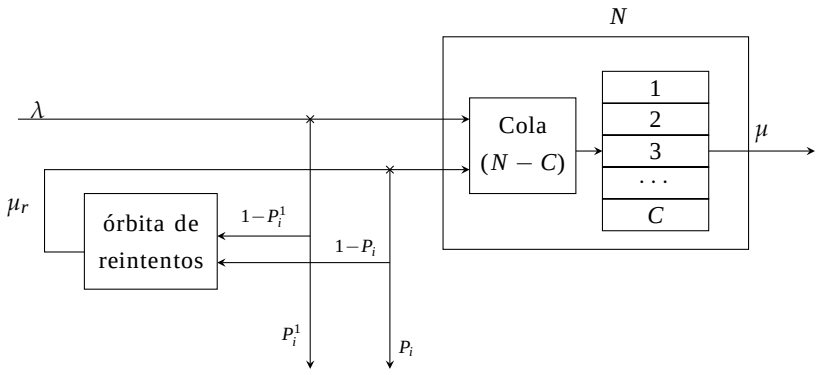


Figura 3.1: Escenario descrito en [GW87] .

sistema. El término a la derecha del signo de igualdad representa la tasa de finalización de servicio. Por otro lado, en (3.4) el primer término identifica la tasa de usuarios primarios bloqueados, mientras que el segundo término la tasa de reintentos bloqueados que vuelve a reintentar. Siendo el término a la derecha de la igualdad la tasa total de reintentos.

Haciendo uso de la suposición RTA podemos identificar:

$$R(k) = \pi(k) \quad \forall k \tag{3.5}$$

Insertando la ecuación (3.5) en (3.3) y teniendo en cuenta la condición de normalización, la probabilidad de bloqueo se obtiene como:

$$\pi(N) = \frac{\frac{(\lambda+M_R)^N}{\mu_1 \ldots \mu_N}}{1 + \frac{\lambda+M_R}{\mu_1} + \ldots + \frac{(\lambda+M_R)^N}{\mu_1 \ldots \mu_N}} \tag{3.6}$$

donde

$$\mu_j = \begin{cases} j\mu & \text{si } j \leq C \\ C\mu & \text{si } j > C \end{cases}$$

También insertando la ecuación (3.4) en (3.5) se puede obtener:

$$\pi(N) = \frac{M_R}{\lambda(1-P_i^1) + M_R(1-P_i)}. \tag{3.7}$$

Finalmente, se observa como las ecuaciones (3.6) y (3.7) ofrecen una ecuación polinómica para M_R de orden $N+1$, que presenta una única solución. Una vez calculada la tasa con la que llegan los reintentos al sistema, M_R, se calcula la probabilidad de bloqueo utilizando (3.6) ó (3.7).

Se observa que la aproximación utilizada en este modelo es independiente de la tasa de reintentos, μ_r. Se intuye que si la tasa de reintentos es elevada, los usuarios reintentarán el acceso a los servidores muy frecuentemente, con lo que la probabilidad de que el sistema se mantenga en congestión será elevada. Sin embargo si la tasa de reintentos es baja, el tiempo entre reintentos aumentará y será más probable que el usuario vea la ocupación media de servidores del sistema. Por lo tanto, se espera que esta aproximación obtenga una buena precisión únicamente cuando la tasa de reintentos es baja.

Capítulo 4

Modelos Truncados

En este capitulo se aborda el estudio de los modelos truncados. Como se comentó en el capítulo 2, estos modelos se caracterizan por reemplazar el espacio de estados infinito original, S^o, por otro de estados finitos, S, resultante de la truncación del espacio original.

En la primera parte de este capítulo se resumen las principales contribuciones que podemos encontrar en la literatura existente, comentando las ventajas e inconvenientes de cada una de ellas. Este análisis servirá de punto de partida para presentar el modelo FM, desarrollado con el fin de mejorar la precisión obtenida con los modelos existentes a costa de un pequeño incremento en el coste computacional.

4.1 Antecedentes

Entre los modelos pertenecientes a este tipo podemos encontrar los trabajos propuestos en [Wil56] y [Ste99]. La solución propuesta por Wilkinson en [Wil56] reemplaza el sistema original, en el que el número de usuarios en la órbita de reintentos no está limitado, por un sistema truncado donde se establece un valor máximo, Q, al número de usuarios reintentando. De este

modo, el espacio de estados resultante tiene la forma:

$$S := \{(k, m) : k \leq C; m \leq Q\}.$$

Este espacio de estados es finito en ambas dimensiones y por tanto, se puede resolver utilizando cualquiera de las técnicas existentes para la resolución de *QBDs* finitos no-homogéneos.

La precisión del modelo aproximado es sensible al valor de Q escogido. Así, la solución será poco precisa si se toman valores de Q pequeños. Mientras que con valores elevados de Q se alcanzará una alta precisión, pero el número de estados del sistema será grande, aumentando el coste computacional. Nos encontraremos una situación parecida en aquellos casos en que el sistema esté muy cargado, puesto que para conseguir una precisión aceptable se requerirá un valor de Q elevado.

Otra solución basada en la truncación del espacio de estados es la propuesta por Stepanov en [Ste99]. Esta solución es algo más compleja que la propuesta por Wilkinson al considerar únicamente aquellos estados cuya probabilidad sea "no despreciable", eliminando aquellos que presenten una probabilidad de estado "despreciable". Este tipo de solución evita, en parte, los problemas mencionados anteriormente. Sin embargo, para cargas elevadas también se requiere un alto coste computacional.

Existen otros modelos truncados que no se limitan a truncar el espacio de estados del sistema sino que, además, lo modifican con el fin de introducir el efecto que tendrían en el sistema los estados eliminados. Entre las propuestas que realizan este tipo de aproximación encontramos la de Fredericks y Reisner en [FR79] para la resolución de un escenario como el reflejado en la figura 4.1. Nótese que este escenario es una simplificación del que encontramos en la figura 3.1 con $P_i^1 = P_i$ y $N = C$. Esta solución reduce el modelo inicial, de un espacio de estados bidimensional y con una dimensión infinita, a un espacio de estados de una única dimensión y finito, en concreto, a una $M/M/C/C$ o sistema de pérdidas Erlang B. Para ello se elimina la dimensión correspondiente al número de usuarios en la órbita de reintentos y se

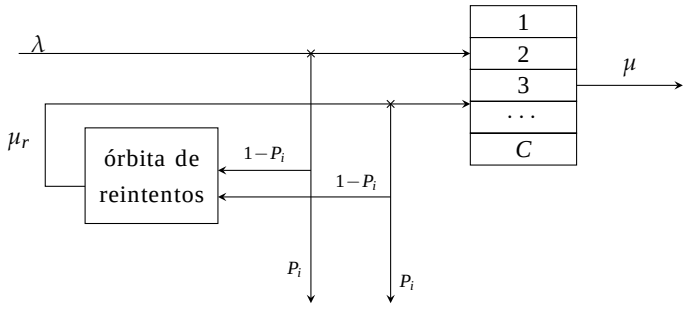


Figura 4.1: Escenario descrito en [FR79] .

introduce su efecto mediante una tasa de llegadas que dependerá del estado del sistema. Esta nueva tasa viene dada por:

$$\lambda(k) = \lambda + \mu_r E(m|k),$$

donde el primer término a la derecha de la igualdad,λ, es la tasa de llegadas de usuarios nuevos y el segundo término, $\mu_r E(m|k)$ representa la tasa de llegadas de los reintentos, donde $E(m|k)$ es número medio de usuarios en la órbita de reintentos, m, condicionado a que hayan k servidores ocupados. De esta forma la tasa total ofrecida al sistema se calcula como:

$$\Omega = \sum_{k=0}^{C} \lambda(k)\pi(k),$$

donde $\pi(k)$ es el porcentaje de tiempo en que hay k servidores ocupados.

Esta tasa puede coincidir o no con el valor deseado. En este último caso será necesario reajustar el valor de $\lambda(k)$, mediante un proceso iterativo. El proceso iterativo utilizado consta de dos partes. La primera parte consiste en una recurrencia *forward*:

1. Damos un valor inicial a $\lambda(0)$

2. Calculamos los valores de $\lambda(1)$ y un parámetro auxiliar $\delta(1)$ a partir de

$\lambda(0)$, así:

$$\delta(1) = 1 + \frac{\mu_r}{\lambda(0)},$$

$$\lambda(1) = \delta(1)\lambda(0) + (1 - \delta(1))\lambda.$$

3. Para $2 \leq j \leq (C-1)$, hacer

$$\delta(j) = \frac{[\mu_r + \lambda(j-1) + (j-1)\mu]\delta(j-1) - (j-1)\mu}{\lambda(j-1)\delta(j-1)}$$

y

$$\lambda(j) = \delta(j)\lambda(j-1) + (1 - \delta(j))\lambda.$$

4. Calculamos $\delta(C)$ como:

$$\delta(C) = \frac{[\mu_r + \lambda(C-1) + (C-1)\mu]\delta(C-1) - (C-1)\mu}{\lambda(C-1)\delta(C-1)},$$

de donde se obtiene

$$\lambda(C) = \frac{1}{1 - \mu_r(1 - P_i)\frac{\delta(C)}{(\mu_r + C\mu)\delta(C) - C\mu}}.$$

Terminada la primera parte se inicia un proceso de recurrencia *backward* en que, haciendo uso de:

$$\lambda(j-1) = \lambda + \frac{\lambda(j) - \lambda}{\delta(j)} \qquad j = C, \ldots, 1$$

se calcula la siguiente iteración de $\lambda(k)$.

El proceso iterativo finalizará cuando el error entre el $\lambda(0)$ calculado en dos iteraciones consecutivas sea menor que un determinado valor ϵ. Calculado $\lambda(k)$, se obtiene el vector de probabilidades de estado haciendo uso de las ecuaciones de balance y la condición de normalización:

$$\pi(k) = \frac{(\frac{\lambda(k)}{\mu})^k / k!}{\sum_{j=0}^{C} (\frac{\lambda(j)}{\mu})^j / j!}. \tag{4.1}$$

de forma que la probabilidad de bloqueo para este modelo pueda calcularse como $P_b = \pi(C)$.

De forma similar al modelo de Fredericks y Reisner [FR79], el modelo propuesto por Marsan et al en [MCL+01] reduce el espacio de estados infinito inicial por otro finito obtenido a partir de la agrupación de todos los niveles del *QBD* en que existen usuarios en la órbita de reintentos. Es decir, se define una variable booleana que indica la presencia ('1') o ausencia ('0') de usuarios reintentando. El espacio de estados resultante viene definido por:

$$S_{Marsan} := \{s = (k, b) : k \leq C; b = 0, 1\},$$

donde k define el número de usuarios siendo servidos, y b es la variable booleana que indica la existencia / ausencia de usuarios reintentando. Sin embargo, esta reducción del espacio de estados tiene un precio. Es fácil observar que este modelo esconde toda la información referente al estado de la órbita de reintentos, puesto que no se sabe cuál es el número de usuarios que se encuentra en la misma. Para compensar este hecho el modelo introduce dos parámetros con el fin de tener en cuenta el efecto de los estados agrupados. Las expresiones de estos parámetros van a depender del escenario estudiado. Aunque en [MCL+01] se consideran diferentes escenarios, en este trabajo y en concreto en la sección 4.2.2 nos vamos a centrar un escenario con población finita.

4.2 *Finite Model*, FM

Los modelos truncados que podemos encontrar en la literatura presentan varios problemas. Así parece evidente que la truncación que hace Wilkinson [Wil56] va a presentar problemas de precisión si deseamos un número de estados pequeño. La búsqueda de precisiones elevadas puede llevar a requerir espacios de estados muy grandes que hagan difícil e incluso imposible la resolución del sistema estudiado. Soluciones como la de Fredericks y Reisner [FR79] o la de Marsan et al [MCL+01], aunque consiguen paliar

este problema y, en general, consiguen buenos resultados en cuanto a precisión, presentan el problema de ofrecer soluciones únicas. En estos casos, dada una configuración del sistema, estos modelos sólo son capaces de ofrecer una solución, que será intrínseca a dicha configuración. De este modo es imposible controlar la precisión que nos ofrece el modelo tal y como ocurría en el modelo de Wilkinson mediante la variación del parámetro Q.

Es por ello que se ha desarrollado, dentro de esta tesis un nuevo modelo, al que denominamos *modelo FM*, que nos permite obtener las ventajas de estos dos tipos de soluciones, por un lado la eficiencia de los modelos como el de Marsan et al y, por otro lado, la posibilidad de ajustar el sistema a resolver para asegurar una determinada precisión.

Con el fin de poder evaluar las prestaciones de este nuevo modelo, se ha propuesto la resolución de un sistema de reintentos con un espacio de estados finito, que permita comparar los resultados de la resolución del modelo original — y exacto— con los resultados obtenidos con alguna de estas aproximaciones. En particular se ha comparado con el modelo propuesto por Marsan et al [MCL+01] y con FM.

4.2.1 Escenario

El escenario estudiado se muestra en la figura 4.2. Se considera una población finita, con lo que el espacio de estados es finito. Existe un colectivo de U usuarios cuyas peticiones son atendidas por C servidores según una ley exponencial de tasa μ. Cuando un usuario solicita servicio por primera vez, y encuentra todos los servidores ocupados, pasa a la órbita de reintentos. Esta órbita consiste en una espera aleatoria —exponencial— de tasa μ_r. Al expirar el tiempo de espera el usuario efectúa un reintento, pudiendo ser exitoso en caso de que encuentre algún servidor libre. De lo contrario, el reintento será fallido y el usuario podrá volver a la órbita de reintentos con probabilidad $(1 - P_i)$ o abandonar el sistema con probabilidad P_i. En cuanto al proceso de

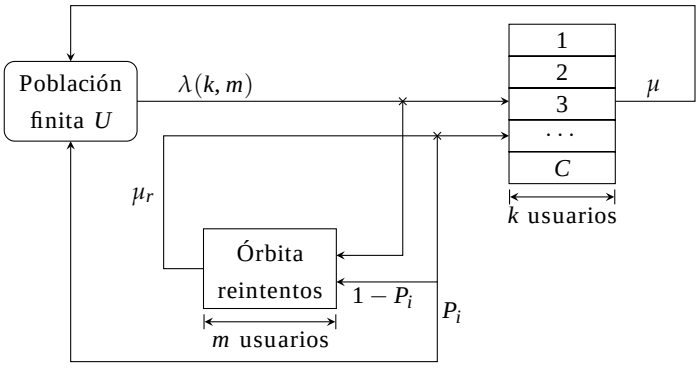


Figura 4.2: Modelo sistema de reintentos con población finita.

llegadas, el tiempo entre llegadas consecutivas es exponencial de tasa

$$\lambda(k,m) = (U-k-m)\gamma,$$

donde γ es la tasa de llegada de un usuario cuando este se encuentra en reposo (*idle*), y k (m) el número de usuarios en servicio (en la órbita de reintentos).

El sistema de reintentos de la figura 4.2 constituye un proceso de Markov cuyo espacio de estados viene definido por

$$S := \{(k,m) : 0 \leq k \leq C; 0 \leq m \leq U-C\}.$$

La figura 4.3 muestra el diagrama de transiciones del sistema y consta de $(U-C+1)\times(C+1)$ estados. El generador infinitesimal, $\mathbf{Q}$, presenta una estructura de *QBD* no-homogéneo, cuyo aspecto es:

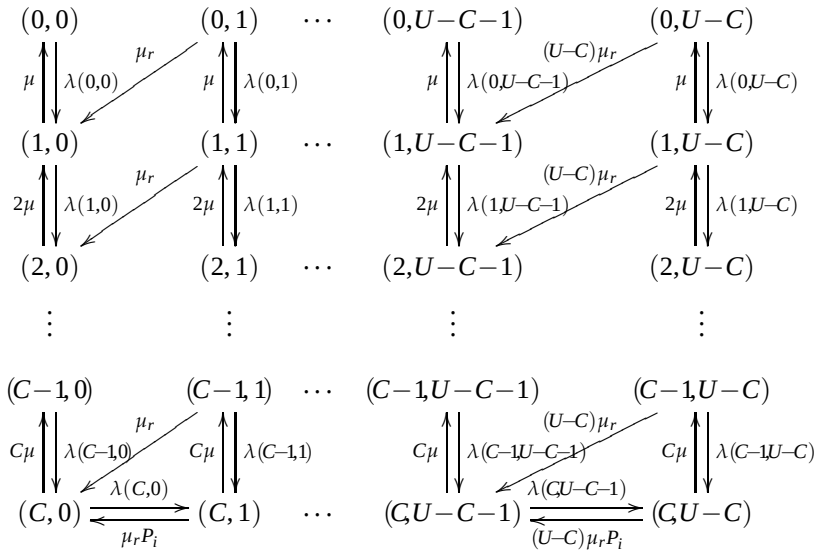


Figura 4.3: Diagrama de transiciones del sistema finito.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & \dots & 0 & 0 \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \dots & 0 & 0 \\ 0 & \mathbf{A}_2^{(2)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_0^{(U-C-2)} & 0 \\ 0 & 0 & \dots & \mathbf{A}_1^{(U-C-1)} & \mathbf{A}_0^{(U-C-1)} \\ 0 & 0 & \dots & \mathbf{A}_2^{(U-C)} & \mathbf{A}_1^{(U-C)} \end{bmatrix}, \tag{4.2}$$

siendo $\mathbf{A}_0^{(m)}$, $\mathbf{A}_1^{(m)}$ y $\mathbf{A}_2^{(m)}$, matrices cuadradas de dimensiones $(C+1)\times(C+$

1) que, en este caso, son de la forma:

$$\mathbf{A_0^{(m)}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \lambda(C,m) \end{bmatrix},$$

$$\mathbf{A_1^{(m)}} = \begin{bmatrix} * & \lambda(0,m) & 0 & \dots & 0 \\ \mu & * & \lambda(1,m) & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda(C-1,m) \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2^{(m)}} = \begin{bmatrix} 0 & m\mu_r & 0 & \dots & 0 \\ 0 & 0 & m\mu_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & m\mu_r \\ 0 & 0 & 0 & \dots & m\mu_r P_i \end{bmatrix}.$$

Los asteriscos que aparecen en $\mathbf{A_1^{(m)}}$ son los valores que hacen que la suma de los elementos de la fila correspondiente del generador $\mathbf{Q}$ sean cero.

Para obtener las probabilidades de estado en régimen permanente, $\boldsymbol{\pi}$ se resuelve $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, junto con la condición de normalización, $\boldsymbol{\pi}\mathbf{e} = 1$, donde $\mathbf{e}$ es un vector columna cuyos elementos son la unidad. Una vez calculado el vector de probabilidades de estado se podrán calcular los diferentes parámetros de mérito del sistema. En concreto, la probabilidad de bloqueo, P_b, se expresa como:

$$P_b = \sum_{m=0}^{U-C} \pi(C,m).$$

Las probabilidades de servicio inmediato, P_{si}, servicio demorado, P_{sd}, y no servicio, P_{ns}, pueden calcularse a partir de las tasas de primeros intentos y reintentos según las relaciones definidas en la ecuación (2.2). Estas tasas, para el escenario considerado, vienen dadas por las expresiones:

$$R_o = \sum_{k=0}^{C} \sum_{m=0}^{U-C} \lambda(k,m)\pi(k,m) \tag{4.3}$$

$$R_{1,s} = \sum_{k=0}^{C-1} \sum_{m=0}^{U-C} \lambda(k,m)\pi(k,m) \tag{4.4}$$

$$R_{1,f} = \sum_{m=0}^{U-C} \lambda(C,m)\pi(C,m) \tag{4.5}$$

$$R_r = \sum_{k=1}^{C} \sum_{m=0}^{U-C} m\mu_r\pi(k,m) \tag{4.6}$$

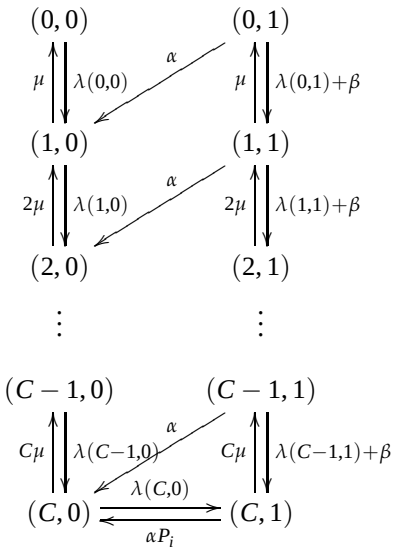
$$R_{r,s} = \sum_{k=1}^{C-1} \sum_{m=0}^{U-C} m\mu_r\pi(k,m) \tag{4.7}$$

$$R_{r,f} = \sum_{m=1}^{U-C} m\mu_r\pi(C,m) \tag{4.8}$$

4.2.2 Modelo aproximado de Marsan et al [MCL+01]

De entre los modelos truncados, se ha elegido el modelo de Marsan et al en [MCL+01] para compararlo con la resolución exacta dado que consigue una muy buena precisión. Este modelo agrega todas las columnas del modelo exacto, a partir de la segunda, en una única columna según se indica en el diagrama de transiciones de la figura 4.4.

La falta de información referente al estado de la órbita, hace necesario considerar aproximaciones para describir dos aspectos: la secuencia de peticiones generada por los usuarios bloqueados y que puede aproximarse por un proceso interrumpido de Poisson (IPP) [Kuc73] de tasa $\overline{m}\mu_r$, con $\overline{m}$ el número medio de usuarios en la órbita de reintentos —suponiendo que ésta no



$$\text{con } \lambda(k,0) = (U-k)\gamma \text{ y } \lambda(k,1) = (U-k-\overline{m})\gamma$$

Figura 4.4: Diagrama de transiciones para el modelo de Marsan et al.

se encuentre vacía—. Además se realiza una aproximación heurística, definiéndose p como la probabilidad de que, tras un reintento exitoso, todavía existan usuarios en la órbita de reintentos. Así, según se observa en la figura 4.4, aparecen dos nuevas contribuciones, $\alpha = \overline{m}\mu_r(1-p)$ y $\beta = \overline{m}\mu_r p$ cumpliéndose que $\alpha + \beta = \overline{m}\mu_r$.

Para calcular los parámetros $\overline{m}$ y p es necesario resolver la ecuación de balance flujos que surge del corte vertical entre los dos niveles del diagrama de transiciones de la figura 4.4 y que se expresa como:

$$\alpha \sum_{k=0}^{C-1} \pi(k,1) + \alpha P_i \pi(C,1) = \lambda(C,0)\pi(C,0). \tag{4.9}$$

Por otro lado, se hace uso de la propiedad de sistemas de colas con incrementos y decrementos unitarios en el tamaño de la población —*crossing up-down argument*—: La tasa de usuarios que abandona la citada órbita y la deja no vacía ha de coincidir con la tasa de peticiones que llegan a la órbita de reintentos y la encuentran no vacía:

$$\sum_{k=0}^{C-1} \beta\pi(k,1) + P_i\beta\pi(C,1) = \lambda(C,1)\pi(C,1). \tag{4.10}$$

Sumando las ecuaciones (4.9) y (4.10) se obtiene:

$$\overline{m}\mu_r\Big[\sum_{k=0}^{C-1} \pi(k,1) + P_i\pi(C,1)\Big] = \lambda(C,0)\pi(C,0) + \lambda(C,1)\pi(C,1). \tag{4.11}$$

De la ecuación (4.10) sabemos que:

$$\sum_{k=0}^{C-1} \pi(k,1) + P_i\pi(C,1) = \frac{\lambda(C,1)\pi(C,1)}{\beta} = \frac{\lambda(C,1)\pi(C,1)}{\overline{m}\mu_r p}. \tag{4.12}$$

Llevando (4.12) a la ecuación (4.11), se consigue:

$$\overline{m}\mu_r \frac{\lambda(C,1)\pi(C,1)}{\overline{m}\mu_r p} = \lambda(C,0)\pi(C,0) + \lambda(C,1)\pi(C,1). \tag{4.13}$$

De donde se despeja p, obteniendo:

$$p = \frac{\lambda(C,1)\pi(C,1)}{(\lambda(C,0)\pi(C,0) + \lambda(C,1)\pi(C,1))}. \tag{4.14}$$

Para obtener la expresión de $\overline{m}$ se utiliza la ecuación (4.9), donde sustituyendo p por la expresión (4.14), se llega a :

$$\overline{m} = \frac{[\lambda(C,0)\pi(C,0) + \lambda(C,1)\pi(C,1)]}{\mu_r[\sum_{k=0}^{C-1} \pi(k,1) + P_i\pi(C,1)]}. \tag{4.15}$$

Es necesario resaltar la dependencia entre los valores de p y $\overline{m}$. Además se observa como las ecuaciones de las probabilidades de estado y las expresiones para la obtención de estos parámetros son mutuamente dependientes. Es

decir, las ecuaciones de balance globales, la ecuación de normalización y las ecuaciones (4.14) y (4.15) forman un sistema de ecuaciones no lineales. Este sistema se resolverá mediante el uso de un proceso iterativo, tal y como se describe a continuación:

1. Tomamos $p = 0$ y $\overline{m} = 1$
2. Calculamos las probabilidades de estado en régimen permanente $\pi(k, b)$
3. Con los resultados obtenidos calculamos los nuevos valores de p y $\overline{m}$ utilizando (4.14) y (4.15)
4. Calculamos el error relativo entre los nuevos y los viejos valores de los parámetros p y $\overline{m}$
   - Si el error obtenido es mayor que un determinado ϵ volvemos al punto (2).
   - En caso contrario, se detiene el proceso iterativo. Se han alcanzado valores aceptables para los nuevos parámetros, alcanzándose asimismo la solución a las probabilidades de estado.

En [MCL+01] se asumía la convergencia de este proceso iterativo. Aunque no ha podido demostrarse la convergencia de éste, los autores evaluaron el modelo en un amplio rango de escenarios con diferentes configuraciones y el proceso ha convergido en todos los casos.

El generador infinitesimal que resulta de este modelo es una cadena de Markov de dos niveles:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} \end{bmatrix}. \tag{4.16}$$

De las cuatro submatrices que componen el generador, únicamente se verán

afectadas por la aproximación $\mathbf{A}_1^{(1)}$ y $\mathbf{A}_2^{(1)}$, que quedarán como:

$$\mathbf{A_1^{(1)}} = \begin{bmatrix} * & \lambda(0,m)+\beta & 0 & \dots & 0 \\ \mu & * & \lambda(1,m)+\beta & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda(C-1,m)+\beta \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2^{(1)}} = \begin{bmatrix} 0 & \alpha & 0 & \dots & 0 \\ 0 & 0 & \alpha & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha \\ 0 & 0 & 0 & \dots & \alpha P_i \end{bmatrix}.$$

Mientras que las otras dos submatrices serán iguales a las que teníamos en el modelo exacto.

A partir de este generador infinitesimal y resolviendo $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, se obtienen las probabilidades de estado. Por otro lado, las tasas como definidas en la ecuación (2.2) se calculan como:

$$R_o = \sum_{k=0}^{C}\sum_{b=0}^{1} \lambda(k,b)\pi(k,b) \tag{4.17}$$

$$R_{1,s} = \sum_{k=0}^{C-1}\sum_{b=0}^{1} \lambda(k,b)\pi(k,b) \tag{4.18}$$

$$R_{1,f} = \sum_{b=0}^{1} \lambda(C,b)\pi(C,b) \tag{4.19}$$

$$R_r = \sum_{k=0}^{C} \overline{m}\mu_r \pi(k,1); \tag{4.20}$$

$$R_{r,s} = \sum_{k=0}^{C-1} \overline{m}\mu_r \pi(k,1); \tag{4.21}$$

$$R_{r,f} = \overline{m}\mu_r \pi(C,1); \tag{4.22}$$

Nótese que las ecuaciones (4.17)-(4.19) y (4.20)-(4.22) son muy similares a las del modelo exacto definidas en las ecuaciones (4.3)-(4.5) y (4.6)-(4.8). Sólo ha sido necesario sustituir los índices del segundo sumatorio: en el modelo exacto correspondían a $0 \leq m \leq U - C$, mientras que en este caso sólo podrá tomar valores 0 ó 1, debido a la agregación de estados. Con estas tasas se obtienen los diferentes parámetros de mérito como:

$$P_{si} = \frac{R_{1,s}}{R_o} \quad ; \quad P_{sd} = \frac{R_{r,s}}{R_o} \quad ; \quad P_{ns} = \frac{R_{ab}}{R_o} = \frac{P_i R_{r,f}}{R_o}$$

Por lo que respecta a la probabilidad de bloqueo —porcentaje de tiempo en que los C servidores están ocupados—, ésta quedará expresada como:

$$P_b = \sum_{b=0}^{1} \pi(C,b).$$

Fijándose en las cuatro probabilidades anteriores, se ha evaluado el error que introduce este modelo aproximado frente a los resultados obtenidos con el modelo exacto. Para ello se considera la configuración descrita en la tabla 4.1.

Nótese que en el caso de fuentes finitas la carga que recibe el sistema depende directamente de las características de dicho sistema. Esto es debido a que la tasa de llegadas instantánea depende del estado del sistema y la proporción de tiempo que el sistema pasa en cada estado depende del número de servidores y el trato que se da a los usuarios bloqueados. En este caso se puede derivar una relación entre la carga ofrecida por fuente en reposo, $\hat{a} = \gamma/\mu$, y la carga total ofrecida prevista, a^*, entendida como la carga que

Tabla 4.1: Valores para los parámetros del sistema finito.

U	120 usuarios
C	30 servidores
μ	1 / 180
l_f $(l_f = \frac{\gamma}{\gamma+\mu})$	0.14 — 0.44
μ_r	0.1
P_i	0.5

las fuentes ofrecerían al sistema si hubiese servidores suficientes para evitar el bloqueo (es decir, $C = U$). Entonces:

$$a^* = U\frac{\hat{a}}{1+\hat{a}} = Ul_f.$$

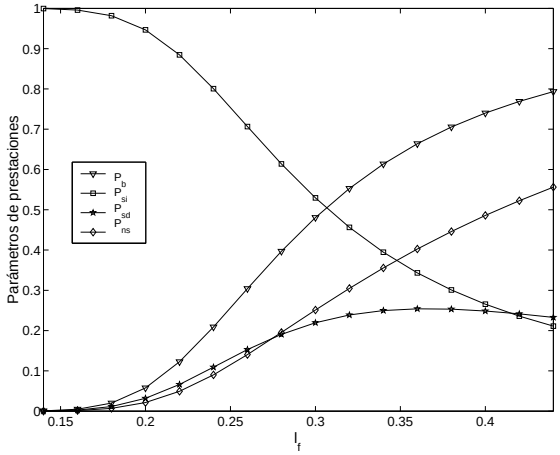
Los resultados para el modelo exacto se muestran en la figura 4.5(a), donde se observa la evolución de los diferentes parámetros de mérito conforme se aumenta la carga por fuente, l_f. Por otra parte, la figura 4.5(b) muestra el error relativo que se produce en los diferentes parámetros de mérito cuando se utiliza el modelo de Marsan. El error relativo se ha calculado como

$$\mid \Gamma^{exact} - \Gamma^{approx} \mid / \Gamma^{exact} \quad \text{donde} \quad \Gamma \in \{P_b, P_{si}, P_{sd}, P_{ns}\}. \tag{4.23}$$

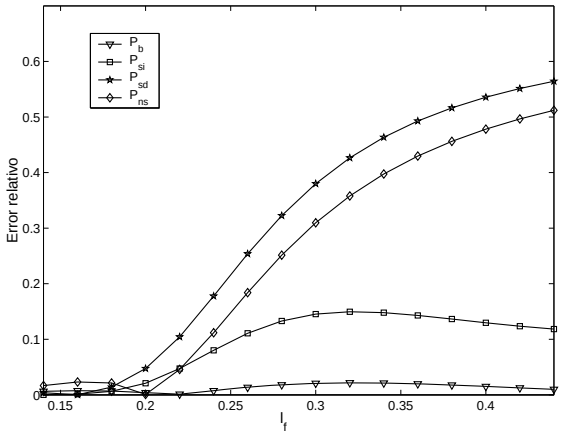
Tal y como se concluye en [MCL+01], el modelo aproximado propuesto en dicho trabajo obtiene muy buenos resultados en términos de probabilidad de bloqueo. Así, se observa que el error relativo respecto al modelo exacto es muy bajo para cualquier carga. Sin embargo, esto no ocurre con el resto de parámetros de merito, esto es, P_{si},P_{sd} y P_{ns}, donde el error relativo puede superar valores del 50 % para cargas muy elevadas.

4.2.3 Desarrollo matemático del modelo FM

El modelo propuesto por Marsan et al en [MCL+01] presenta ciertos problemas de precisión, como acabamos de observar. Asimismo, el uso del modelo



(a) Resultados del modelo exacto.



(b) Error relativo obtenido con el modelo de Marsan et al.

Figura 4.5: Evaluación del modelo de Marsan et al.

exacto no siempre va a ser posible, puesto que en muchos casos nos encontraremos con sistemas de población infinita y por tanto con un sistema con un espacio de estados también infinito y heterogéneo. Incluso en el caso de contar con población finita podemos encontrar sistemas con un espacio de estados extremadamente grande que haga que la resolución del sistema sea muy costosa en términos de memoria y uso de CPU, o incluso imposible de resolver. Es por ello que se plantea la necesidad de otros modelos que ofrezcan una buena aproximación no sólo para la probabilidad de bloqueo sino también para el resto de parámetros de interés.

El modelo propuesto puede entenderse como una generalización del modelo presentado en [MCL+01]. Mientras que en dicho modelo se agregaban todos los niveles en que existían usuarios en la órbita de reintentos en un único nivel, en el modelo propuesto se agregan sólo aquellos niveles en que el número de usuarios en la órbita de reintentos sea igual o superior a un determinado valor Q. El sistema resultante constará de $Q+1$ niveles, concretamente $m = \{0,\ldots,Q\}$, donde los niveles del 0 al $Q-1$ corresponden con los del modelo exacto, mientras que el nivel Q incluyen el efecto de todos los niveles en que hay Q o más usuarios en la órbita de reintentos. Así pues, el modelo propuesto es un modelo parametrizable, es decir, Q no tiene un valor fijo, sino que dependerá de los objetivos que se pretendan alcanzar. Así, un valor alto de Q nos permitirá obtener una alta precisión en los resultados obtenidos, mientras que un valor bajo de Q nos asegurará un coste computacional bajo. El objetivo del modelo es llegar a un compromiso entre precisión y coste computacional. Nótese que, si tomamos $Q = 1$, el modelo aproximado resultante coincidirá con el modelo propuesto en [MCL+01], mientras que si se toma $Q = U - C$ tendremos el modelo exacto.

El espacio de estados de la cadena de Markov resultante tiene la forma:

$$S := \{(k,m) : 0 \leq k \leq C; 0 \leq m \leq Q\}.$$

donde el conjunto de estados (k,Q) para $0 \leq k \leq C$ corresponde con la situación en que existen Q o más usuarios en la órbita de reintentos.

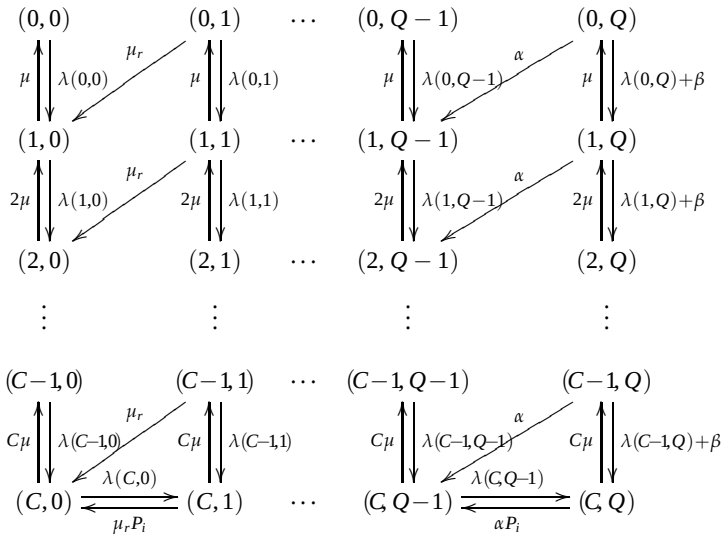


Figura 4.6: Diagrama de transiciones del modelo FM.

La figura 4.6 muestra el diagrama de transiciones del modelo propuesto. Las primeras $Q-1$ columnas son idénticas a las que teníamos en el modelo exacto —figura 4.3—. La columna Q muestra la aproximación realizada mediante la agregación de estados. Así, igual que en el Modelo presentado en [MCL+01] van a aparecer los parámetros $\overline{m}$ y p. El parámetro $\overline{m}$ representa el número de usuarios reintentando cuando existen Q o más usuarios en la órbita de reintentos. Mientras que el parámetro p representa la probabilidad de que, tras un reintento exitoso, el número de usuarios en la órbita de reintentos se mantenga igual o superior a Q. Obviamente, $(1-p)$ representará la probabilidad de que tras un reintento exitoso en la órbita queden menos de Q usuarios, condicionado a que inicialmente había Q o más usuarios en la órbita. De este modo, la tasa de reintentos en los estados (k,Q) se divide en dos contribuciones, α y β. La primera contribución, α, corresponde a las transiciones de (k,Q) hasta $(k+1,Q-1)$, pudiendo aproximar dicha tasa como $\alpha = \overline{m}\mu_r(1-p)$. La segunda contribución corresponde con las transiciones

desde (k, Q) hasta $(k+1, Q)$ y se aproxima como $\beta = \overline{m}\mu_r p$.

Para resolver el sistema de ecuaciones vamos a recurrir al generador infinitesimal que define este modelo. Este generador presenta la misma estructura que el generador del modelo exacto, pero con Q niveles frente a los $U - C$ del modelo exacto.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & \dots & 0 & 0 \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \dots & 0 & 0 \\ 0 & \mathbf{A}_2^{(2)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_0^{(Q-2)} & 0 \\ 0 & 0 & \dots & \mathbf{A}_1(Q-1) & \mathbf{A}_0^{(Q-1)} \\ 0 & 0 & \dots & \mathbf{A}_2^{Q} & \mathbf{A}_1^{Q} \end{bmatrix}.$$

Las submatrices de dicho generador infinitesimal son las mismas que aparecían en el modelo exacto para los niveles $m \leq Q-1$, mientras que para el último nivel, $m = Q$, se observarán ciertas diferencias debida a la aproximación realizada. Dichos cambios se darán en las submatrices $\mathbf{A_1^{(Q)}}$ y $\mathbf{A_2^{(Q)}}$, pero no en $\mathbf{A_0^{(Q)}}$ puesto que éste sólo depende de las tasas de llegada de los usuarios y por tanto permanecerá igual a las matrices del modelo exacto, así:

$$\mathbf{A_1^{(Q)}} = \begin{bmatrix} * & \lambda(0,\overline{m}) + \beta & 0 & \dots & 0 \\ \mu & * & \lambda(1,\overline{m}) + \beta & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda(C-1,\overline{m}) + \beta \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A}_2^{(Q)} = \begin{bmatrix} 0 & \alpha & 0 & \dots & 0 & 0 \\ 0 & 0 & \alpha & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \alpha \\ 0 & 0 & 0 & \dots & 0 & \alpha P_i \end{bmatrix}.$$

Estas submatrices se definen igual que las de nivel 1 del modelo propuesto en [MCL+01], con la diferencia de que la definición de los parámetros α y β no es la misma. Esta característica resulta lógica puesto que el modelo propuesto lo que hace es desplazar la agrupación de estados que se realizaba en el modelo propuesto por Marsan et al hasta un nivel tal que permita asegurar un determinado nivel de precisión.

También será necesario realizar ciertos cambios en las expresiones de las tasas a calcular.

$$\begin{aligned} R_o &= \sum_{k=0}^{C}\sum_{m=0}^{Q}\lambda(k,m)\pi(k,m) \\ R_{1,s} &= \sum_{k=0}^{C-1}\sum_{m=0}^{Q}\lambda(k,m)\pi(k,m) \\ R_{1,f} &= \sum_{m=0}^{Q}\lambda(C,m)\pi(C,m) \end{aligned}$$

$$\begin{aligned} R_r &= \sum_{k=0}^{C}\sum_{m=0}^{Q-1} m\mu_r\pi(k,m) + \overline{m}\mu_r\sum_{k=0}^{C}\pi(k,Q), \\ R_{r,s} &= \sum_{k=0}^{C-1}\sum_{m=0}^{Q-1} m\mu_r\pi(k,m) + \overline{m}\mu_r\sum_{k=0}^{C-1}\pi(k,Q), \\ R_{r,f} &= \sum_{m=0}^{Q-1} m\mu_r\pi(C,m) + \overline{m}\mu_r\pi(C,Q). \end{aligned}$$

Con todo ello, los parámetros p y $\overline{m}$ se pueden estimar mediante el balance del flujo de probabilidades que cruza cada uno de los cortes verticales del

diagrama de transiciones. Este proceso genera las siguientes ecuaciones:

$$\begin{aligned}
&\lambda(C,0)\pi(C,0) = \mu_r \sum_{k=0}^{C-1} \pi(k,1) + \mu_r P_i \pi(C,1),\\
&\lambda(C,1)\pi(C,1) = 2\mu_r \sum_{k=0}^{C-1} \pi(k,2) + 2\mu_r P_i \pi(C,2),\\
&\dots\\
&\lambda(C,Q-1)\pi(C,Q-1) = \overline{m}\mu_r(1-p) \sum_{k=0}^{C-1} \pi(k,Q) + \overline{m}\mu_r(1-p)P_i\pi(C,Q).
\end{aligned} \tag{4.24}$$

Sumando todas estas ecuaciones, se obtiene:

$$\begin{aligned}
\sum_{m=0}^{Q-1} \lambda(C,m)\pi(C,m) &= \sum_{m=1}^{Q-1} m\mu_r \sum_{k=0}^{C-1} \pi(k,m) + P_i \sum_{m=1}^{Q-1} m\mu_r \pi(C,m)\\
&+ \overline{m}\mu_r(1-p) \sum_{k=0}^{C-1} \pi(k,Q) + \overline{m}\mu_r(1-p)P_i\pi(C,Q).
\end{aligned}$$

Si reorganizamos esta última ecuación, tenemos:

$$\begin{aligned}
\left[R_{1,f} - \lambda(C,Q)\pi(C,Q)\right] = \left[R_{r,s} - \overline{m}\mu_r \sum_{k=0}^{C-1} \pi(k,Q)\right] + \left[R_{ab} - \overline{m}\mu_r P_i \pi(C,Q)\right] +\\
+ \overline{m}\mu_r(1-p) \sum_{k=0}^{C-1} \pi(k,Q) + \overline{m}\mu_r(1-p)P_i\pi(C,Q).
\end{aligned}$$

Dado que, por definición, la tasa de primeros intentos fallidos es igual a la tasa de reintentos exitosos más la tasa de abandonos, es decir, $R_{1,f} = R_{r,s} + R_{ab}$ podemos obtener:

$$\lambda(C,Q)\pi(C,Q) = \overline{m}\mu_r p \sum_{k=0}^{C} \pi(k,Q) - \overline{m}\mu_r p(1-P_i)\pi(C,Q). \tag{4.25}$$

A partir de la última ecuación de (4.24) y (4.25) se obtiene:

$$p = \frac{\lambda(C,Q)\pi(C,Q)}{\lambda(C,Q-1)\pi(C,Q-1) + \lambda(C,Q)\pi(C,Q)}. \tag{4.26}$$

$$\overline{m} = \frac{\lambda(C, Q-1)\pi(C, Q-1) + \lambda(C, Q)\pi(C, Q)}{\mu_r[\sum_{k=0}^{k=C-1} \pi(k, Q) + P_i\pi(C, Q)]}.$$

Al igual que ocurría en el modelo de Marsan et al [MCL+01], y como vemos en las ecuaciones de p y $\overline{m}$, existe una dependencia entre las probabilidades de estado y estos dos parámetros. Por tanto, es necesario seguir un proceso iterativo como el descrito para calcular estos parámetros. La principal diferencia con el proceso iterativo descrito para Marsan es que en este caso los valores iniciales serán $p = 0$ y $\overline{m} = Q$. Con estos valores se calculan las probabilidades de estado en régimen permanente, $\pi(k, m)$ con $0 \leq k \leq C$ y $0 \leq m \leq Q$. Partiendo de estas probabilidades, se calculan los valores de la siguiente iteración de p y $\overline{m}$. El proceso se repite hasta que la precisión relativa, ϵ, se encuentre por debajo de un determinado valor, en nuestro caso se toma por defecto un valor de $\epsilon = 10^{-4}$.

4.2.4 Evaluación numérica

En esta sección se comparan los resultados obtenidos con el modelo FM frente a los obtenidos con el modelo propuesto por Marsan et al en [MCL+01]. Además se evalúa la reducción en el coste computacional de estos dos modelos frente al coste que tendría resolver el modelo exacto.

Para realizar éste estudio comparativo se hace uso de los mismos valores para los parámetros del sistema que en el caso anterior —valores de la tabla 4.1— en que se comparaba el modelo de Marsan con la resolución exacta.

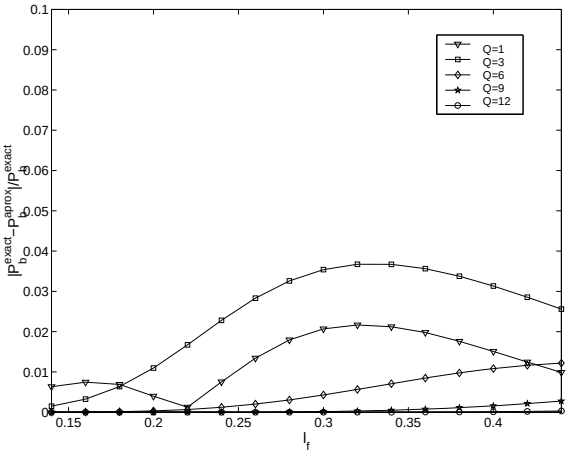
Estudio de la precisión

Un primer punto de interés en el análisis del modelo propuesto es el estudio del error cometido por el modelo FM conforme varía Q. Recordemos que el estudio del modelo FM nos ofrece también los resultados que obtendríamos con el modelo de Marsan et al si se toma $Q = 1$, y por tanto, estos resultados también pueden servir para comparar ambos modelos. Se ha evaluado el

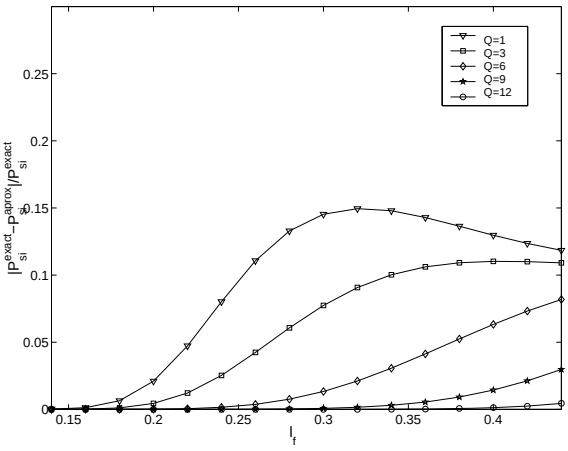
error relativo —que viene dado por la ecuación (4.23)— para los diferentes parámetro de mérito, P_b, P_{si}, P_{sd}, P_{ns}. El estudio se ha realizado para diferentes cargas de fuente y diferentes valores de Q.

Las figuras 4.7 y 4.8 muestran los resultados de este análisis. Para una curva dada, en general, conforme aumenta la carga de fuente, aumenta el error en el modelo aproximado. Este efecto se observa más claramente en las figuras 4.8(a) y 4.8(b) que corresponden con la probabilidad de servicio demorado y de no servicio. Este comportamiento resulta lógico puesto que un aumento en la carga supone un aumento en el número de usuarios en la órbita de reintentos. Desgraciadamente, el modelo aproximado no va a poder reproducir esa ocupación de órbita con total precisión, al ser en este punto del modelo en el que ha realizado la simplificación. El mayor interés de estas gráficas se encuentra en la diferencia en los resultados obtenidos según el valor de Q utilizado. En el caso de la probabilidad de bloqueo, figura 4.7(a), el uso de $Q = 1$ consigue mejores resultados que otros valores de Q superiores. En el resto de parámetros de mérito, para una carga dada, el error relativo disminuye conforme aumenta el valor de Q, es decir, conforme la aproximación se acerca al modelo exacto. Esta singularidad en la probabilidad de bloqueo se debe a que el cálculo de este parámetro depende únicamente de las probabilidades de estado, mientras que el resto de parámetros depden-den de forma directa de los parámetros de la aproximación y por tanto del valor de Q escogido. De este mismo modo, seleccionar un valor de μ_r mayor hará que este efecto disminuya o incluso llegue a desaparecer puesto que se disminuirá el número medio de usuarios reintentando y con ello, el error que se puede producir en los parámetros de la aproximación.

Además, el valor de Q que garantiza una buena precisión depende de la carga, l_f. A mayor carga necesitaremos una Q mayor para garantizar una determinada precisión. Sin embargo, el valor de Q necesario va a ser mucho menor que $U - C$, que constituye el caso exacto. Por ejemplo, para el peor escenario estudiado, $l_f = 0.44$, con $Q = 12$ se consigue un error relativo menor que 10^{-2} en todos los parámetros de mérito, reduciendo de este modo en un 85 % el número de estados. Para el caso de un sistema con una carga

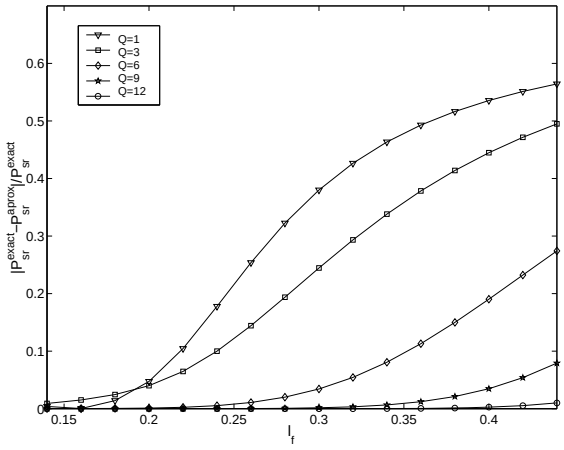


(a) Probabilidad de bloqueo, P_b

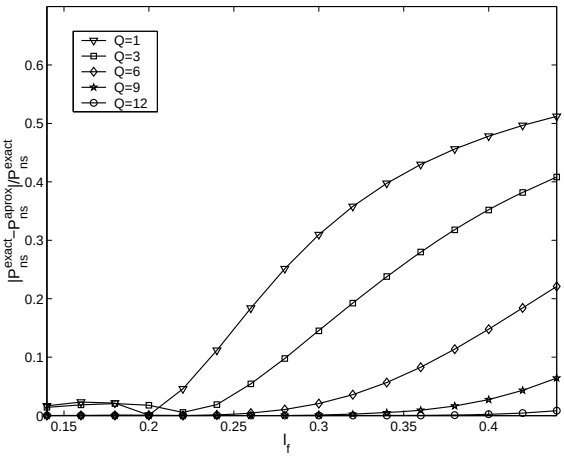


(b) Probabilidad de servicio inmediato, P_{si}

Figura 4.7: FM: Error relativo en P_b y P_{si}.



(a) Probabilidad de servicio demorado, P_{sd}



(b) Probabilidad de no servicio, P_{ns}

Figura 4.8: FM: Error relativo en P_{sd} y P_{ns}.

media, como podría ser para $l_f = 0.22$ —con $P_b = 0.12$—, la reducción del espacio de estados es del 99.95 % para $Q = 1$ —modelo de Marsan et al— y del 99.50 % para $Q = 10$, con respecto al modelo exacto. Obviamente, puesto que el escenario está menos cargado, la reducción en el espacio de estados es todavía mayor. Si nos fijamos en el aumento de complejidad —número de estados— resultante de usar el modelo FM, en vez del modelo de Marsan et al [MCL+01], se puede observar que no hay mucha variación.

Tras el estudio de diferentes configuraciones del sistema se concluye, como norma general, que un valor de Q del orden de $Q \simeq 0.15(U - C)$, independientemente de la carga por fuente, garantiza una buena precisión en todos los casos. En este sentido, se ha estudiado lo que ocurre cuando se utilizan otras configuraciones del sistema. Por ejemplo, la tabla 4.2 muestra los resultados para los diferentes parámetros de prestaciones cuando variamos la población del sistema, U, en el caso de $l_f = 0.22$ y manteniendo fijos el resto de parámetros. Obviamente, el grado de servicio obtenido por el sistema varía con la población: a mayor población, la calidad de servicio obtenida por los usuarios empeora.

Tabla 4.2: Probabilidades de servicio al variar la población del sistema

(U, Q)	P_{si}(exacto)	P_{si} (FM)	P_{sd}(exacto)	P_{sd} (FM)	P_{ns}(exacto)	P_{ns} (FM)
$(61, 2)$	0.999998	0.999998	$1.451 \cdot 10^{-6}$	$1.433 \cdot 10^{-6}$	$5.884 \cdot 10^{-7}$	$5.992 \cdot 10^{-7}$
$(77, 3)$	0.999575	0.999566	$2.870 \cdot 10^{-4}$	$2.854 \cdot 10^{-4}$	$1.384 \cdot 10^{-4}$	$1.396 \cdot 10^{-4}$
$(91, 5)$	0.992881	0.992862	$4.569 \cdot 10^{-3}$	$4.564 \cdot 10^{-3}$	$2.550 \cdot 10^{-3}$	$2.553 \cdot 10^{-3}$
$(107, 6)$	0.949919	0.949817	$3.015 \cdot 10^{-2}$	$3.012 \cdot 10^{-2}$	$1.993 \cdot 10^{-2}$	$1.994 \cdot 10^{-4}$

El valor Q que asegura un error despreciable respecto al valor exacto de las probabilidades de servicio se encuentra, en todos los casos, por debajo del 15 % del valor máximo que puede tomar.

Se concluye que, para las situaciones en que la presencia de reintentos está garantizada, es decir, cargas medias-altas, el modelo FM consigue mejores resultados que el modelo de Marsan et al [MCL+01]. Aunque el modelo de Marsan et al consigue muy buenos resultados en términos de precisión para la probabilidad de bloqueo, esto no es cierto para el resto de parámetros

de mérito. Por tanto, es mejor recurrir al modelo FM que permite asegurar buenos resultados en todos los parámetros con unos valores de Q bajos. Además el modelo FM presenta la posibilidad de ajustarse a las necesidades de modelado y mejorar la precisión mediante el incremento del valor de Q.

Coste Computacional

En esta sección se estudia el modelo FM en términos de coste computacional. Este coste computacional se puede medir en operaciones de coma flotante, *flops*[1].

El estudio realizado se centra en dos aspectos complementarios. El primer aspecto a tener en cuenta es la reducción en el coste computacional que resulta de resolver el modelo aproximado en lugar del modelo exacto. El segundo aspecto considerado es la metodología de resolución utilizada. Según el algoritmo utilizado conseguiremos una mayor o menor reducción en el coste computacional. Con el fin de hacer frente a estos dos aspectos, se han estudiado diferentes algoritmos, en concreto el algoritmo GJL propuesto por Gaver, Jacobs y Latouche en [GJL84] y los algoritmos propuestos por Servi en [Ser02]. En el apéndice C.2 se muestras las características de estos dos algoritmos.

Todo el estudio se ha realizado para el escenario anterior con $l_f = 0.22$, que es un caso típico de carga media, y por tanto, asegura la existencia de reintentos. Nótese que, en este escenario, el uso de una $Q = 6$ sería suficiente para garantizar una buena precisión, tal y como se observa en las figuras 4.7-4.8.

La figura 4.9 muestra los resultados obtenidos. En ella se observa el coste computacional de ambos algoritmos crece conforme aumenta el valor de Q

[1]Los resultados numéricos, así como los costes computaciones asociados se han obtenido usando Matlab. Este producto permite definir el coste computacional de un algoritmo en términos de *flops* computados de la siguiente forma: sumas y restas suponen 1 *flop* si los operadores son reales y 2 *flops* si se trata de número complejos. Productos y divisiones requieren 1 *flop* si el resultado es real y 6 *flops* en el caso de ser complejo.

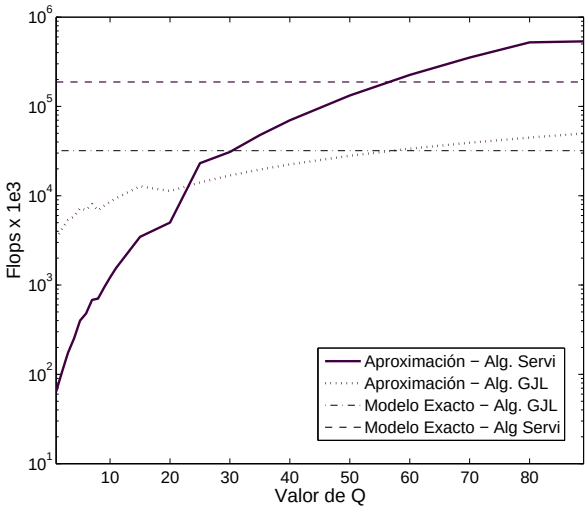


Figura 4.9: Coste computacional para diferentes algoritmos.

puesto que el sistema a resolver presenta un mayor número de estados. Sin embargo, si comparamos la evolución de ambos algoritmos, el coste computacional del algoritmo de Servi crece con mayor rapidez que el coste del algoritmo GJL, que presenta una pendiente menor. Por otro lado, destacar que el algoritmo de Servi presenta costes menores que el algoritmo GJL para valores bajos de Q. Concretamente, hasta llegar a valores de $Q = 25$ el algoritmo de Servi es mejor que el GJL, mientras que a partir de esta Q el algoritmo GJL presenta un coste menor. Puesto que los valores de Q que aseguran una precisión aceptable son del orden de $Q = 6$ es recomendable el uso del algoritmo de Servi en esta situación. En la figura 4.9 se muestra también el coste de resolver el modelo exacto con ambos algoritmos. Puede resultar extraño que conforme el valor de Q aumenta y se acerca al valor exacto, $Q = U - C$, la resolución del modelo FM resulta más costosa, con cualquiera de los algoritmos, que la resolución del modelo exacto. Sin embargo, esto es lógico si se tiene en cuenta que, mientras que el modelo exacto sólo resuelve el QBD aso-

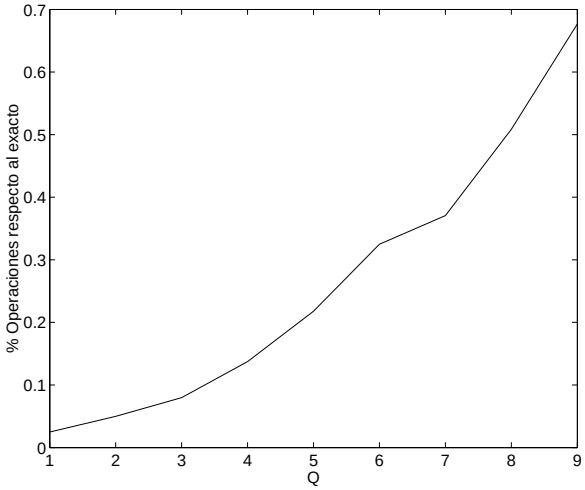


Figura 4.10: Alg. Servi: coste del modelo aproximado frente al exacto.

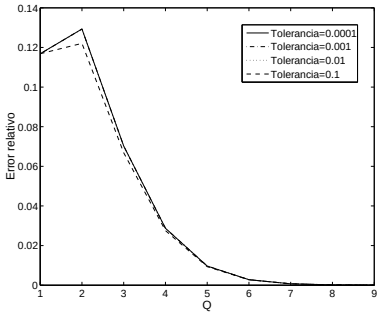
ciado al modelo, en el caso del modelo aproximado, a parte de la resolución del sistema, existe una sobrecarga de computo debida al proceso iterativo asociado al cálculo de los parámetros $\overline{m}$ y p.

Si nos centramos en el algoritmo de Servi, se aprecia una reducción del número de operaciones en torno a los dos órdenes de magnitud —para los valores de Q de interés— respecto a la resolución del modelo exacto. Esta reducción en el coste se observa con detalle en la figura 4.10. Esta figura muestra el número de operaciones necesarias para resolver el sistema para diferentes valores de Q, con respecto a las operaciones necesarias para resolver el modelo exacto. Nótese que ambos modelos se han resuelto haciendo uso del algoritmo de Servi. Como se observa en la figura, se pueden conseguir reducciones del 99.6 % respecto al modelo exacto para una $Q = 6$ que es un valor suficiente para asegurar una buena precisión. Si comparamos este coste con el requerido para resolver el modelo con $Q = 1$ —modelo de Marsan et al [MCL+01]—, obviamente el número de operaciones requerido por

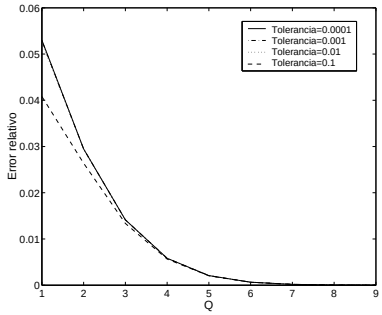
el modelo FM es mayor que el requerido por el modelo de Marsan et al. Se pasa de unas 70000 operaciones cuando se usa el modelo de Marsan et al a las 450000 del modelo FM con $Q = 6$. Esto supone un incremento considerable, pero que en ambos casos es mucho menor que el coste de resolver el modelo exacto.

Se puede conseguir una mayor reducción en el coste computacional mediante el uso de una precisión menor en la estimación de los parámetros $\overline{m}$ y p. Si hasta el momento se ha utilizado una tolerancia de $\epsilon = 10^{-4}$ para el cálculo de estos parámetros, se pueden usar tolerancias menores, $\epsilon = \{10^{-3}, 10^{-2}, 10^{-1}\}$, con el fin de reducir el coste computacional. Obviamente, reducir la tolerancia conlleva una reducción en el número de iteraciones necesarias para llegar a la solución. Sin embargo, esta reducción en la precisión conducirá a un mayor error en los parámetros de prestaciones calculados. La figura 4.11 muestra los errores relativos en las probabilidades de servicio para el escenario con $l_f = 0.22$ y diferentes valores de tolerancia. Tal y como se observa en dicha figura, la reducción de la tolerancia no va a afectar sustancialmente al error relativo que se produce en los parámetros de mérito. De forma que, una $Q = 6$ sigue garantiza un buen funcionamiento del sistema, ya que el error que se produce en los parámetros de mérito al usar una tolerancia u otra es despreciable. Obviamente, para $Q > 6$, usar una tolerancia u otra en el proceso iterativo no supone diferencia alguna en los resultados obtenidos, mientras que para $Q < 6$ sí que existe cierta discrepancia. En el caso peor, para la probabilidad de no servicio —fig. 4.11(c)—, utilizar una tolerancia menor puede llevar a un incremento de la Q mínima necesaria para garantizar cierta precisión. Por ejemplo, si se quiere asegurar un error relativo menor que 10^{-2}, para el caso de una tolerancia de 10^{-4} con una $Q = 2$ sería suficiente, mientras que con una tolerancia de 10^{-1} sería necesario una $Q = 3$. Esta variación se diluye en el resto de probabilidades de servicio, donde las Q necesarias van a ser mayores, y la diferencia entre usar una tolerancia u otra es insignificante y, en ningún caso, afecta al valor de Q.

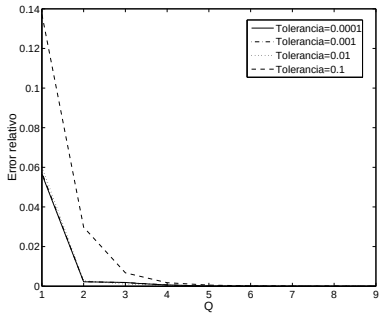
Se concluye que el factor determinante de este modelo FM es Q y, en gene-



(a) Probabilidad de servicio demorado, P_{sd}



(b) Probabilidad de servicio inmediato, P_{si}



(c) Probabilidad de no servicio, P_{ns}

Figura 4.11: Error en las probabilidades de servicio respecto a la tolerancia.

ral, se pueden usar tolerancias pequeñas en el proceso iterativo. Este hecho tiene la ventaja adicional de reducir el coste computacional. La figura 4.12 muestra la diferencia de coste que se consigue utilizando diferentes tolerancias. Así, para valores entorno a $Q = 6$ el coste computacional obtenido con tolerancias bajas se reduce por un factor 2 respecto a los casos con tolerancias elevadas.

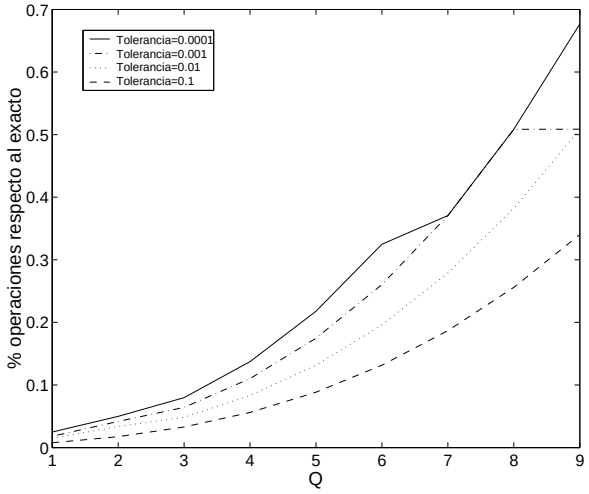


Figura 4.12: Coste computacional para diferentes tolerancias.

4.2.5 Conclusiones

El modelo FM mejora sustancialmente la precisión del modelo de Marsan et al propuesto en [MCL+01], consiguiendo una buena aproximación para todos los parámetros de mérito a tener en cuenta en un sistema de reintentos, a cambio de un pequeño incremento del coste computacional. Además el modelo presenta la posibilidad de ajustar el parámetro Q con el fin de conseguir un compromiso entre precisión y coste computacional.

La reducción de la complejidad del sistema a resolver es considerable, pudiendo observarse que con espacios de estados un 85 % más pequeños que el del modelo exacto se consiguen buenas precisiones para todos los parámetros. Se concluye que la reducción del espacio de estados que lleva a cabo el modelo FM junto la utilización del algoritmo de Servi y la posibilidad de utilizar tolerancias bajas en el proceso iterativo para calcular los parámetros de la aproximación, permiten una reducción del coste computacional aproxi-

madamente del 99 % respecto al modelo exacto.

Capítulo 5

Modelos truncados generalizados

Si el valor a partir del que truncamos, Q, es suficientemente grande, se puede utilizar cualquiera de los modelos truncados descritos en el capítulo anterior para aproximar el modelo original, puesto que asegurará una precisión elevada. Sin embargo, en determinados casos —escenarios con cargas muy elevadas, etc.— pueden ser necesarios valores muy elevados de Q para garantizar la precisión objetivo, con lo que el coste computacional requerido para resolver el sistema sería muy elevado o incluso imposible de resolver con este tipo de modelos. Por tanto, en dichas circunstancias, es conveniente usar otro tipo de modelos aproximados que resulten más eficientes.

En este apartado se estudia un tercer tipo de modelo aproximado denominado modelos truncados generalizados —*generalized truncated models*— que, en general, van a obtener mejores resultados que los modelos truncados. Estos modelos reemplazan el espacio de estados infinito inicial por otro también infinito, pero con determinadas características que hagan factible la obtención de las probabilidades de estado del sistema en régimen permanente.

En la literatura se pueden distinguir dos formas de aproximar un sistema de reintentos mediante el uso de modelos truncados generalizados. La primera de ellas consiste en considerar que la tasa de reintentos es infinita en determinadas circunstancias. Esta suposición permite eliminar ciertos estados, de

forma que el espacio de estados resultante presente ciertas características que hagan posible calcular las probabilidades de estado en régimen permanente. Esta solución es la que se observa en [Fal83] y [AP02]. Otras aproximaciones, como la presentada en [NR90], se basan en la homogeneización del espacio de estados a partir de un determinado nivel. Esta homogeneización permitirá el uso de los diferentes algoritmos desarrollados para resolver *QBDs* infinitos y homogéneos.

En este capítulo presentamos los principales modelos truncados generalizados que se pueden encontrar en la literatura. Hecho esto, las secciones 5.2 y 5.3 presentan varios modelos propuestos con el fin de mejorar los modelos existentes. En concreto, la sección 5.2 presenta un modelo basado en la eliminación de estados, mientras que la sección 5.3 presenta varios modelos de homogeneización. Por último se comparan los resultados de los modelos desarrollados, estudiando los modelos tanto en términos de complejidad, entendida como número de estados necesario para conseguir una determinada precisión, como en términos de tiempo de resolución.

5.1 Antecedentes

5.1.1 Modelo de Falin [Fal83]

Entre las soluciones que pertenecen a este tipo de modelo, la más sencilla es la presentada por Falin en [Fal83] y que, tal y como se demuestra en [AA02], converge al valor exacto. Esta solución está basada en hacer que tasa de reintentos sea infinita a partir de un determinado número de usuarios en la órbita de reintentos, Q. En dichos casos, el servicio de los usuarios en la órbita deja de ser el propio de una órbita y se convierte en el comportamiento típico de una cola de espera. Es decir, a partir del nivel Q, el sistema se convierte en una cola $M/M/1$ con tasa de llegada λ y tasa de servicio $C\mu$. La tasa de reintentos de este sistema se representa, por tanto, como una función con dos

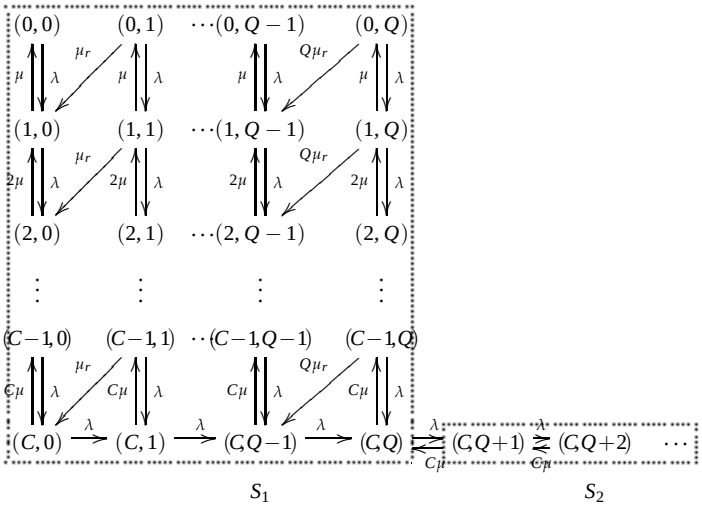


Figura 5.1: Diagrama de transiciones del modelo de Falin.

intervalos:

$$\mu_r(m) = \begin{cases} m\mu_r & \text{si } m \leq Q \\ \infty & \text{si } m > Q \end{cases}$$

El diagrama de transiciones de este modelo se muestra en la figura 5.1. Resaltar que este modelo no considera la posibilidad de tener usuarios impacientes que abandonen el sistema sin haber obtenido servicio.

Para resolver el sistema resultante aprovecharemos las características del diagrama de transiciones que se descompone en dos partes bien diferenciadas. Por una parte tendríamos el subespacio que queda delimitado por el recuadro de la izquierda de la figura 5.1, y que viene definido como:

$$S_1 := \{(k,m) : k \leq C; m \leq Q\}.$$

Por otra parte aparece un subespacio S_2, que corresponde con el espacio de estados de una cola $M/M/1$ y que, por tanto, se define como:

$$S_2 := \{(k,m) : k = C; m > Q\}.$$

Para obtener las probabilidades de estado del espacio total, primero se obtienen las probabilidades de estado del subespacio S_1, para lo que se puede usar cualquiera de los métodos existentes para la resolución de *QBDs* no-homogéneos y finitos [GJL84, Ser02]. A continuación se obtienen las probabilidades de estado del subespacio correspondiente a la cola $M/M/1$. Para ello se parte de la probabilidad de estado $\pi(C,Q)$, ya calculada durante la primera fase. Es decir, si consideramos que el generador infinitesimal del subespacio S_1 viene dado por:

$$\mathbf{Q}_{S_1} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & \dots & 0 & 0 \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \dots & 0 & 0 \\ 0 & \mathbf{A}_2^{(2)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_0^{(Q-2)} & 0 \\ 0 & 0 & \dots & \mathbf{A}_1^{(Q-1)} & \mathbf{A}_0^{(Q-1)} \\ 0 & 0 & \dots & \mathbf{A}_2^{(Q)} & \mathbf{A}_1^{(Q)} \end{bmatrix}, \tag{5.1}$$

donde las matrices $\mathbf{A}_0^{(m)}$, $\mathbf{A}_1^{(m)}$ y $\mathbf{A}_2^{(m)}$, con $0 \leq m \leq Q$, son matrices cuadradas de dimensiones $(C+1)\times(C+1)$ que, en este caso, son de la forma:

$$\mathbf{A_0^{(m)}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \lambda \end{bmatrix},$$

$$\mathbf{A_1^{(m)}} = \begin{bmatrix} * & \lambda & 0 & \dots & 0 \\ \mu & * & \lambda & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2^{(m)}} = \begin{bmatrix} 0 & m\mu_r & 0 & \dots & 0 \\ 0 & 0 & m\mu_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & m\mu_r \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Los asteriscos que aparecen en la diagonal principal de $\mathbf{A_1^{(m)}}$ son valores que hacen que la suma de los elementos de la fila correspondiente del generador infinitesimal $\mathbf{Q}$ sea cero.

Podemos obtener el vector de probabilidades de estado resolviendo $\boldsymbol{\pi Q} = \mathbf{0}$, junto con la condición de normalización. Entre las probabilidades de estado calculadas tendremos $\pi(C, Q)$, a partir de la cual calculamos las probabilidades del subespacio S_2 como:

$$\pi(C, Q+i) = \left(\frac{\lambda}{C\mu}\right)^i \pi(C, Q). \tag{5.2}$$

Obtenidas las probabilidades de estado se calcularán los distintos parámetros de prestaciones de interés. Por ejemplo para calcular la probabilidad de bloqueo tenemos:

$$P_b = \sum_{m=0}^{\infty} \pi(C, m). \tag{5.3}$$

Este tipo de solución reduce considerablemente el valor de Q a partir del cual se trunca el espacio de estados original para conseguir una determinada precisión. Así, el valor de Q que utiliza este modelo es mucho menor que el utilizado, por ejemplo, por el modelo de truncación básico definido por Wilkinson y explicado en el capítulo anterior.

5.1.2 Modelo de Artalejo y Pozo [AP02]

Más recientemente, Artalejo y Pozo [AP02] han propuesto un modelo conceptualmente similar al propuesto por Falin en [Fal83]. Aprovechando el hecho de que la cola $M/M/2$ con reintentos presenta solución exacta [Han87,

Art96], Artalejo y Pozo extienden en [AP02] el modelo de Falin [Fal83]. Así, si el modelo de Falin reduce el sistema a una $M/M/1$ a partir de un nivel Q, el modelo de Artalejo y Falin lo reduce a una $M/M/2$. Obviamente, este modelo resulta más complejo de resolver que el modelo propuesto por Falin, pero ofrece la ventaja de ser más eficiente, en el sentido de que el sistema a resolver para conseguir una determinada precisión, considera un valor de Q menor.

En este modelo, la tasa de reintentos no depende únicamente del número de usuarios en la órbita de reintentos, sino que también es dependiente del número de servidores ocupados. La función que define esta tasa viene dada por:

$$\mu_r(m) = \begin{cases} \infty & \text{si } 0 \leq k \leq C-2 \text{ y } m \geq Q+1 \\ m\mu_r & \text{en cualquier otro caso} \end{cases}$$

donde destaca el hecho de que la tasa de reintentos depende de la segunda dimensión, es decir de m, siendo por tanto no-homogénea, a diferencia de lo que ocurría en el modelo de Falin. Nótese también que, al igual que ocurría con el modelo de Falin, este modelo tampoco permite la existencia de usuarios impacientes en la órbita de reintentos. Con todo esto, el diagrama de transiciones que define este modelo se muestra en la figura 5.2.

Las probabilidades de estado de este sistema se pueden obtener como solución de las ecuaciones $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$. Para este sistema dichas ecuaciones se pueden agrupar en dos tipos, las correspondientes a los estados en que $m \leq Q$ y aquellas correspondientes a la zona simplificada del modelo, es decir a los niveles $m > Q$. Los estados del nivel $m = Q$ constituyen la frontera entre ambas zonas.

Los estados de la primera zona pueden agruparse en varios intervalos. Así cuando $k < C$ y $m < Q$ tenemos la siguiente ecuación de flujos:

$$(\lambda + m\mu_r + k\mu)\pi(k,m) = \lambda\pi(k-1,m) + (k+1)\mu\pi(k+1,m) + (m+1)\mu_r\pi(k-1,m+1)$$

y para la última fila del diagrama, es decir, para los estados $k = C$ y $m < Q$:

$$(\lambda + C\mu)\pi(C,m) = \lambda\pi(C-1,m) + \lambda\pi(C,m-1) + (m+1)\mu_r\pi(C-1,m+1).$$

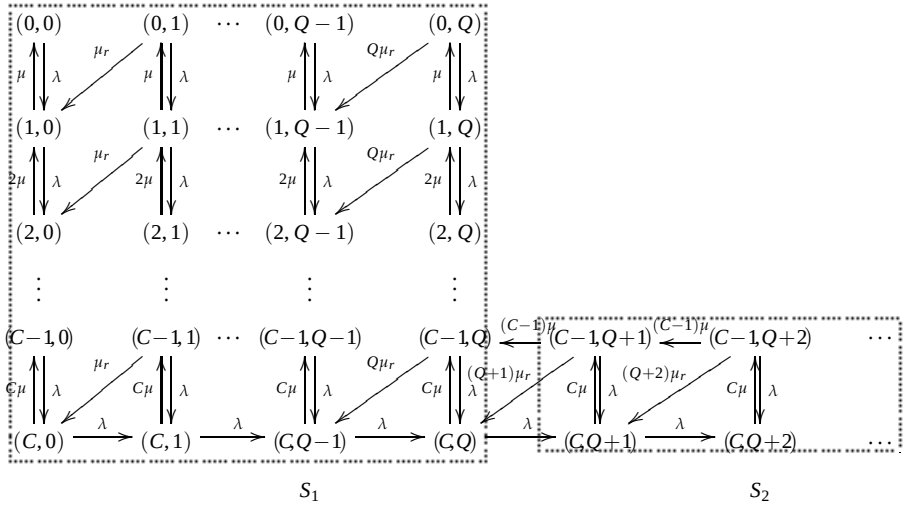


Figura 5.2: Diagrama de transiciones del modelo de Artalejo y Pozo [AP02]

Las ecuaciones de balance del nivel $m = Q$ son:

$$(\lambda + Q\mu_r + k\mu)\pi(k,Q) = \lambda\pi(k-1,Q) + (k+1)\mu\pi(k+1,Q) \qquad 0 \le k \le C-2.$$

La ecuación de balance para el estado $\pi(C-1,Q)$ es:

$$(\lambda+Q\mu_r+(C-1)\mu)\pi(C-1,Q) = \lambda\pi(C-2,Q)+C\mu\pi(C,Q)+(C-1)\mu\pi(C-1,Q+1). \tag{5.4}$$

Por otra parte, haciendo uso de la ecuación de balance de los flujos de entrada-salida entre los niveles $m = Q$ y $m = Q+1$:

$$\pi(C-1,Q+1)[(C-1)\mu + (Q+1)\mu_r] = \lambda\pi(C,Q), \tag{5.5}$$

Insertando la ecuación (5.5) en (5.4) tenemos:

$$[\lambda+(C-1)\mu+Q\mu_r]\pi(C-1,Q) = \lambda\pi(C-2,Q)+[C\mu+\frac{(C-1)\mu\lambda}{(C-1)\mu+(Q+1)\mu_r}]\pi(C,Q).$$

De forma similar para $\pi(C,Q)$ se obtiene:

$$[(\lambda+C\mu)(C-1)\mu+C\mu(Q+1)\mu_r]\pi(C,Q)=\lambda[(C-1)\mu+(Q+1)\mu_r][\pi(C,Q-1)+\pi(C-1,Q)].$$

Por otro lado, las ecuaciones para el intervalo simplificado del espacio, es decir, la parte correspondiente a la cola $M/M/2$ de estados, son:

$$\pi(C-1,m)[(C-1)\mu+m\mu_r]=\lambda\pi(C,m-1) \qquad m\geq Q+1,$$

que es la forma generalizada de la ecuación (5.5). Por otro lado, las ecuaciones correspondientes al balance de los flujos entrantes y salientes de los estados $\pi(C-1,m)$ y $\pi(C,m)$ con $m\geq Q+1$, pueden expresarse, respectivamente, como:

$$[\lambda+(C-1)\mu+m\mu_r]\pi(C-1,m)=(C-1)\mu\pi(C-1,m+1)+C\mu\pi(C,m) \quad m\geq Q+1$$

$$[\lambda+C\mu]\pi(C,m)=\lambda\pi(C,m-1)+\lambda\pi(C-1,m)+(m+1)\mu_r\pi(C-1,m+1) \quad m\geq Q+1$$

Con estas tres ecuaciones es posible expresar $\pi(C-1,m)$ y $\pi(C,m)$, para $m\geq Q+1$ a partir de una probabilidad de estado conocida, como es $\pi(C,Q)$:

$$\pi(C-1,m)=\pi(C,Q)\left(\frac{\lambda}{C\mu}\right)^{m-Q}\frac{C\mu}{(C-1)\mu+(Q+1)\mu_r}\prod_{n=Q+1}^{m-1}\frac{\alpha+n}{\beta+n} \quad m\geq Q+1 \tag{5.6}$$

$$\pi(C,m)=\pi(C,Q)\left(\frac{\lambda}{C\mu}\right)^{m-Q}\frac{(C-1)\mu+(m+1)\mu_r}{(C-1)\mu+(Q+1)\mu_r}\prod_{n=Q+1}^{m}\frac{\alpha+n}{\beta+n} \quad m\geq Q+1 \tag{5.7}$$

con $\alpha=(\lambda+(C-1)\mu)/mu_r$ y $\beta=1+(C-1)(\lambda+C\mu)/(C\mu)$.

De este modo, se pueden calcular todas las probabilidades de estado haciendo uso de un algoritmo del tipo *forward elimination, backward substitution* como el que se indica en [AP02] y que consta de los siguientes pasos:

1. Inicializar el vector de probabilidades de estado con el valor $\pi(0,Q)=1$
2. Se toma $m=Q$ y calculamos:

$$\pi(k,Q)=\frac{(\lambda+(k-1)\mu+Q\mu_r)\pi(k-1,Q)-\lambda\pi(k-2,Q)}{k\mu} \quad 1\leq k\leq C-1$$

$$\pi(C,Q) = \frac{(\lambda + (C-1)\mu + Q\mu_r)\pi(C-1,Q) - \lambda\pi(C-2,Q)}{C\mu + (\lambda(C-1)\mu)/((C-1)\mu + (Q+1)\mu_r)}$$

Nótese que $\pi(k,m)$ será cero para aquellas duplas (k,m) que no formen parte del espacio de estados definido en este modelo.

3. Tomar $m = m-1$ y calcular las probabilidades de estado de este nivel. Para ello se calculará $\pi(C,m)$ como:

$$\pi(C,m) = \frac{(m+1)\mu_r}{\lambda} \sum_{i=0}^{C-1} \pi(i,m+1) \quad 0 \leq m \leq Q-1$$

y, calculando recursivamente el resto de probabilidades de estado de la siguiente forma:

$$\pi(k,m) = \frac{D_{k,m} - \gamma_{k,m}\pi(k+1,m)}{b_{k,m} + \beta_{0,m}} \quad 0 \leq k \leq C-1$$

donde

$$\begin{aligned}
\alpha &= -\lambda \\
\beta_{k,m} &= \lambda + k\mu + m\mu_r \\
\gamma_{k,m} &= -(k+1)\mu \\
\delta_{k,m} &= (m+1)\mu_r \pi_{k-1,m+1} \\
b_{0,m} &= 0, \qquad b_{k,m} = \frac{k\mu(b_{k-1,m} + m\mu_r)}{b_{k-1,m} + \beta_{0,m}} \quad 1 \leq k \leq C-1 \\
D_{0,m} &= 0, \qquad D_{k,m} = \delta_{k,m} - \frac{\alpha D_{k-1,m}}{b_{k-1,m} + \beta_{0,m}} \quad 1 \leq k \leq C-1
\end{aligned}$$

4. Repetir el paso anterior hasta que $m = 0$.

Terminado este proceso se calculan las probabilidades de estado de la zona simplificada, es decir, $S = \{(k,m) : k = C-1, C; m \geq Q+1\}$, a partir de las ecuaciones (5.6) y (5.7). Este proceso puede realizarse de forma más sencilla haciendo uso de funciones hypergeométricas. Obtenido el vector de probabilidades de estado, es posible calcular cualquiera de los parámetros del sistema. Así, por definición, la probabilidad de bloqueo se obtiene como:

$$P_b = \sum_{m=0}^{\infty} \pi(C,m). \tag{5.8}$$

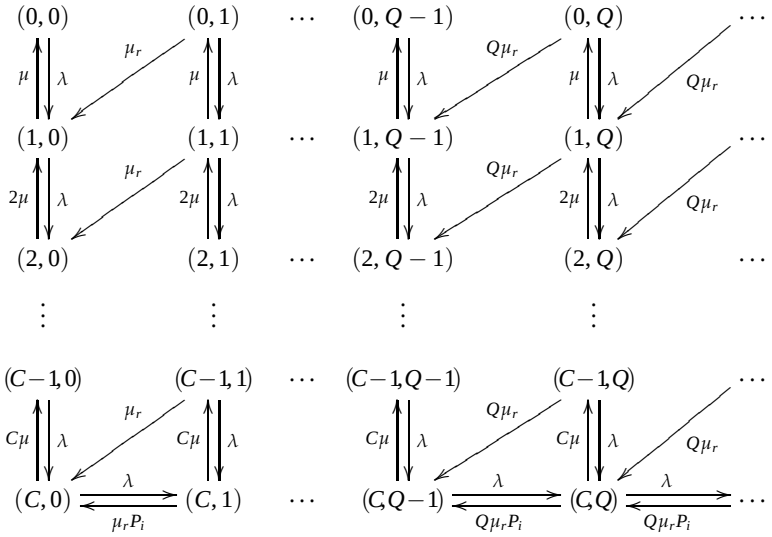


Figura 5.3: Diagrama de transiciones del modelo de Neuts y Rao

5.1.3 Modelo de Neuts y Rao [NR90]

A diferencia de los modelos de Falin, otros modelos optan por realizar una homogenización del espacio de estados que permita una resolución sencilla del sistema. Este es el caso del modelo propuesto por Neuts y Rao en [NR90]. Este modelo se basa en fijar la tasa de reintentos a un valor de $Q\mu_r$ a partir de un determinado nivel Q. Así, la tasa de reintentos en este modelo es de la forma:

$$\mu_r(m) = \begin{cases} m\mu_r & \text{si } m < Q \\ Q\mu_r & \text{si } m \geq Q \end{cases}$$

Apareciendo, de este modo, un diagrama de transiciones como el que podemos observar en la figura 5.3.

Otra característica importante que presenta este modelo es el hecho de permitir que los usuarios abandonen el sistema sin ser servidos, es decir, permite la existencia de usuarios impacientes. En la figura 5.3 se observa

la existencia de una probabilidad de abandono del sistema, P_i, cuando el usuario encuentra todos los servidores ocupados.

El modelo aproximado resultante se resolverá mediante una aproximación denominada *matrix-geometric* propuesta por Neuts en [Neu81]. Para ello, se parte del generador infinitesimal del modelo, que presenta la siguiente forma:

$$\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & 0 & \dots & 0 & 0 \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \mathbf{A}_0^{(1)} & \dots & 0 & \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A}_1^{(Q-1)} & \mathbf{A}_0^{(Q-1)} \\ 0 & 0 & 0 & \dots & \mathbf{A}_2 & \mathbf{A}_1 + \mathbf{R}\mathbf{A}_2 \end{bmatrix},$$

donde $\mathbf{R}$ es la *rate matrix*, definida como la única solución no negativa de la ecuación cuadrática:

$$\mathbf{R}^2\mathbf{A}_2 + \mathbf{R}\mathbf{A}_1 + \mathbf{A}_0 = 0.$$

La matriz $\mathbf{R}$ suele calcularse mediante el siguiente proceso iterativo:

$$\mathbf{R}(n+1) = -\left(\mathbf{A}_0 + \mathbf{R}^2(n)\mathbf{A}_2\right)\mathbf{A}_1^{-1}, \tag{5.9}$$

utilizando como iteración inicial $\mathbf{R}(0) = \mathbf{0}$ y parando dicho proceso cuando el error entre dos iteraciones consecutivas es menor que un determinado error ϵ, es decir, $\max_{i,j} |R_{ij}(n) - R_{ij}(n-1)| < \epsilon$. Este método iterativo es el más sencillo, pero existen otros procesos iterativos más eficientes como es el propuesto en [LR99, Section 8.4].

Calculada la matriz $\mathbf{R}$, el vector de probabilidades de estado para los casos en que $m \leq Q$ se obtiene resolviendo:

$$[\pi_0 \dots \pi_Q]\hat{\mathbf{Q}} = \mathbf{0},$$

combinada con la condición de normalización:

$$\sum_{l=0}^{Q-1} \pi_l \mathbf{e} + \pi_Q (\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = 1.$$

donde $\boldsymbol{\pi}_m = [\pi(0,m),\ldots,\pi(C,m)]$.

Como $\hat{\mathbf{Q}}$ es una matriz finita, este sistema puede resolverse utilizando cualquiera de los métodos definidos en la literatura [GJL84, YL94, Ser02]. Para obtener la probabilidad de los estados en la parte homogénea del modelo, que no aparece en $\hat{\mathbf{Q}}$, se hace uso de la siguiente expresión:

$$\boldsymbol{\pi}_{Q+n} = \boldsymbol{\pi}_Q \mathbf{R}^n$$

En [AA02] se comprueba la convergencia de este modelo al exacto.

5.2 LM: modelo de limitación del espacio de estados

Como ejemplos de modelos basados en la reducción del espacio de estados tenemos el modelo de Falin [Fal83] y el de Artalejo y Pozo [AP02]. La diferencia entre estos dos modelos radica en que, mientras que el modelo de Falin resulta muy sencillo de resolver, el modelo de Artalejo y Pozo tiene una complejidad matemática adicional, apareciendo varios procesos recursivos que pueden llegar a hacer costosa la resolución. Sin embargo, si observamos el espacio de estados necesario para conseguir una determinada precisión en los diferentes parámetros de mérito, el modelo de Artalejo y Pozo consigue la precisión deseada a partir de un valor de Q mucho menor que el que se necesitaría si se utilizase el modelo de Falin.

En esta sección se presenta, como contribución propia de esta tesis, un modelo que, partiendo del modelo de Artalejo y Pozo, reduce el coste computacional del mismo sin necesidad de incrementar considerablemente el valor de Q necesario para garantizar una precisión objetivo. Este modelo se ha denominado *modelo LM*.

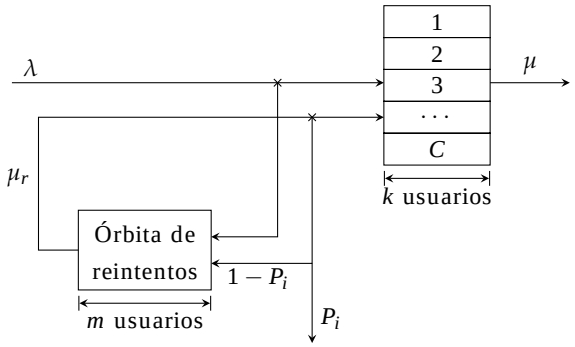


Figura 5.4: Modelo del sistema infinito.

5.2.1 Escenario

Para el estudio de este modelo, así como de los modelos de homogenización que se presentan en secciones posteriores se ha considerado un escenario de población infinita como el descrito en en el capítulo 2 y que se muestra también en la figura 5.4.

En este escenario, los usuarios que llegan al sistema tratan de acceder a uno de los C servidores del sistema. El proceso de llegada se modela como un proceso de Poisson de tasa λ, y se considera que cada usuario hace uso de un único servidor. El tiempo de servicio se considera exponencialmente distribuido con tasa μ. Cuando una nueva petición encuentra todos los servidores ocupados, pasa a la órbita de reintentos, cuya capacidad se considera infinita. Tras un tiempo distribuido exponencialmente y de tasa μ_r, el usuario reintentará el acceso al sistema. Si encuentra algún servidor libre se considera que es un reintento exitoso pues consigue ser servido. Por contra, si encuentra todos los servidores ocupados, el reintento será fallido y el usuario puede, bien dejar la órbita de reintentos con probabilidad P_i, bien volver a la órbita de reintentos para tratar de acceder tras cierto tiempo con la probabilidad complementaria, $(1 - P_i)$. Este modelo se puede representar como un CTMC

de dimensiones:

$$\mathcal{S} := \{(k,m) : k \leq C; m \in \mathbb{Z}_+\}$$

y por tanto constituye un *QBD* infinito y no-homogéneo.

5.2.2 Modelo LM

El modelo LM agrupa el par de estados $\{(C-1,m),(C,m)\}$ para $m \geq Q$ del modelo de Artalejo y Pozo [AP02] en un único estado. Nótese que puesto que el modelo LM deriva del modelo de Artalejo y Pozo tampoco considera la posibilidad de que un usuario abandone el sistema sin ser servido, es decir, $P_i = 0$. Las tasas de entrada y salida del nuevo estado se calculan a partir de las tasas de entrada y salida de los diferentes estados que forman la agrupación y por tanto, se mantiene la heterogeneidad. El diagrama de transiciones se observa en la figura 5.5, donde los estados encuadrados representan los estados resultantes de dicha agrupación. En la figura 5.6 se muestra qué estados se han agrupado y cómo.

La idea es agrupar los dos estados del nivel m, el $(C-1,m)$ y el (C,m) en un único estado, con la intención de reducir complejidad computacional. De esta forma el modelo resultante se asemejaría al de Falin, pero con tasas variables. Es claro pensar que el estado resultante de la agregación se caracterizará por un tiempo de residencia exponencial, cuyo valor medio lo identificaremos con el tiempo medio de residencia en el nivel m original. También deberemos identificar las tasas de salida hacia el estado agregado $m+1$ y hacia el estado agregado $m-1$. Para esta segunda fase, tendremos el cuenta las probabilidades de abandonar el nivel m original, previo a la agregación, hacia el nivel superior $m+1$ y hacia el nivel inferior $m-1$.

En primer lugar, fijándonos en un nivel genérico m, con $m > Q$, observamos que el tiempo de residencia de los dos estados del nivel m, el $(C-1,m)$ y el (C,m), se rigen por sendas leyes exponenciales, con funciones generatrices

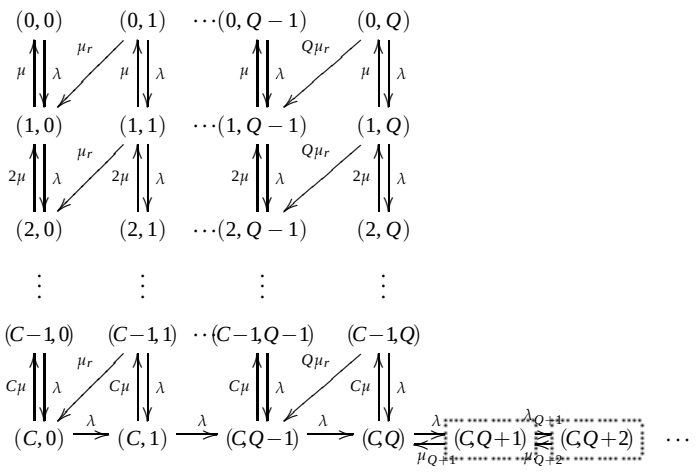


Figura 5.5: Diagrama de transiciones del modelo Agrupado

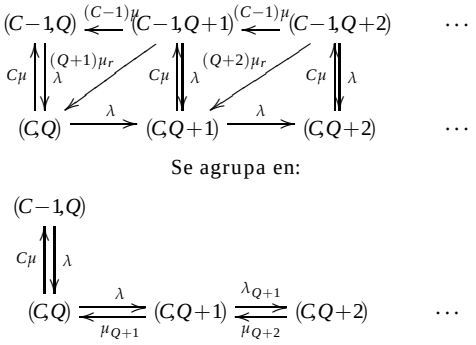


Figura 5.6: Agrupación de estados

dadas por, respectivamente:

$$f^*_{C-1,m}(s) = \frac{\lambda + m\mu_r + (C-1)\mu}{s + \lambda + m\mu_r + (C-1)\mu}; f^*_{C,m}(s) = \frac{\lambda + C\mu}{s + \lambda + C\mu} \qquad (5.10)$$

De la figura 5.6 tenemos las siguientes probabilidad de transición

$$
\begin{array}{lll}
(C-1,m) \to (C,m) & \text{con prob.} & p_{(C-1,m)(C,m)} = \frac{\lambda}{\lambda+m\mu_r+(C-1)\mu} \\
(C-1,m) \to (C,m-1) & \text{con prob.} & p_{(C-1,m)(C,m-1)} = \frac{m\mu_r}{\lambda+m\mu_r+(C-1)\mu} \\
(C-1,m) \to (C-1,m-1) & \text{con prob.} & p_{(C-1,m)(C-1,m-1)} = \frac{(C-1)\mu}{\lambda+m\mu_r+(C-1)\mu} \\
(C,m) \to (C,m+1) & \text{con prob.} & p_{(C,m)(C,m+1)} = \frac{\lambda}{\lambda+C\mu} \\
(C,m) \to (C-1,m) & \text{con prob.} & p_{(C,m)(C-1,m)} = \frac{C\mu}{\lambda+C\mu}
\end{array}
$$

A continuación definimos la transformada de Laplace de los tiempos de estancia en el nivel m condicionados a que se inicia la vista en un determinado estado, el $(C-1,m)$ ó el (C,m) y se abandona el citado nivel m por otro determinado estado, el $(C-1,m)$ ó el (C,m). Denotemos tales transformadas por $f^*_{(C-1,m)(C-1,m)}(s)$, $f^*_{(C-1,m)(C,m)}(s)$, $f^*_{(C,m)(C-1,m)}(s)$ y $f^*_{(C,m)(C,m)}(s)$ éstas vienen dadas por:

$$
\begin{aligned}
f^*_{(C-1,m)(C-1,m)}(s) &= \frac{f^*_{C-1,m}(s)[p_{(C-1,m)(C-1,m-1)}+p_{(C-1,m)(C,m-1)}]}{1-f^*_{C-1,m}(s)p_{(C-1,m)(C,m)}f^*_{C,m}(s)p_{(C,m)(C-1,m)}} \\
f^*_{(C-1,m)(C,m)}(s) &= \frac{f^*_{C-1,m}(s)p_{(C-1,m)(C,m)}f^*_{C,m}(s)p_{(C,m)(C,m+1)}}{1-f^*_{C-1,m}(s)p_{(C-1,m)(C,m)}f^*_{C,m}(s)p_{(C,m)(C-1,m)}} \\
f^*_{(C,m)(C-1,m)}(s) &= \frac{f^*_{C,m}(s)p_{(C,m)(C-1,m)}f^*_{C-1,m}(s)[p_{(C-1,m)(C-1,m-1)}+p_{(C-1,m)(C,m-1)}]}{1-f^*_{C-1,m}(s)p_{(C-1,m)(C,m)}f^*_{C,m}(s)p_{(C,m)(C-1,m)}} \\
f^*_{(C,m)(C,m)}(s) &= \frac{f^*_{C,m}(s)p_{(C,m)(C,m+1)}}{1-f^*_{C-1,m}(s)p_{(C-1,m)(C,m)}f^*_{C,m}(s)p_{(C,m)(C-1,m)}}
\end{aligned}
\tag{5.11}
$$

Con las anteriores ecuaciones definimos la transformada de Laplace del tiempo de estancia en el nivel M condicionado a que dicha estancia da comienzo en el estado $(C-1,m)$, respectivamente (C,m), esto es

$$
\begin{aligned}
f^*_{(C-1,m)*}(s) &= f^*_{(C-1,m)(C-1,m)}(s) + f^*_{(C-1,m)(C,m)}(s) \\
f^*_{(C,m)*}(s) &= f^*_{(C,m)(C-1,m)}(s) + f^*_{(C,m)(C,m)}(s)
\end{aligned}
\tag{5.12}
$$

De (5.12) es inmediato comprobar que $f^*_{(C-1,m)*}(0) = 1$ y $f^*_{(C,m)*}(0) = 1$.

Una ponderación lineal de las dos expresiones de (5.12) nos dará definitivamente la transformada de Laplace del tiempo de residencia o estancia en el nivel m. Dicha ponderación se obtiene según sigue. El nivel m puede alcanzarse vía el estado $(C-1,m)$ o vía el estado (C,m). Sea $v_{(C-1,m)}$, respectivamente $v_{(C,m)}$, la probabilidad de iniciar una visita al nivel m vía el estado $(C-1,m)$, respectivamente vía el estado (C,m). Dichas probabilidades se obtienen al resolver la cadena de Markov homogénea $\mathbf{v}_m = [v_{(C-1,m)}, v_{(C,m)}] = \mathbf{v}_m\mathbf{\Pi}$, esto es

$$\begin{aligned}
&\mathbf{v}_m = [v_{(C-1,m)}, v_{(C,m)}] = \mathbf{v}_m\mathbf{\Pi} = \mathbf{v}_m \begin{pmatrix} \pi_{1,1} & \pi_{1,2} \\ \pi_{2,1} & \pi_{2,2} \end{pmatrix} \\
&\pi_{1,1} = f^*_{(C-1,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C-1,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}} \\
&\pi_{1,2} = f^*_{(C-1,m)(C-1,m)}(0) + f^*_{(C-1,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}} \\
&\pi_{2,1} = f^*_{(C,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C-1,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}} \\
&\pi_{2,2} = f^*_{(C,m)(C-1,m)}(0) + f^*_{(C,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}}
\end{aligned} \tag{5.13}$$

Resulta inmediato verificar que la matriz $\mathbf{\Pi}$ es estocástica. Resolviendo el sistema anterior obtendremos $\mathbf{v}_m = [v_{(C-1,m)}, v_{(C,m)}]$:

$$\mathbf{v}_m = [v_{(C-1,m)}, v_{(C,m)}] = \left[\frac{\pi_{2,1}}{\pi_{1,2}+\pi_{2,1}}, \frac{\pi_{1,2}}{\pi_{1,2}+\pi_{2,1}}\right]$$

y consecuentemente la TL de la distribución del tiempo de estancia en el nivel m esto es:

$$r^*_m(s) = v_{(C-1,m)}f^*_{(C-1,m)*}(s) + , v_{(C,m)}f^*_{(C,m)*}(s) \tag{5.14}$$

El tiempo medio de residencia en el nivel m, nos viene dado por la primera derivada de la ecuación (5.14), el cual tomaremos como tiempo medio de

residencia o estancia en el estado agregado:

$$\lambda_m + \mu_m = -\frac{1}{r_m^{*\prime}(0)} = -\frac{1}{\nu_{(C-1,m)} f_{(C-1,m)*}^{*\prime}(0) + \nu_{(C,m)} f_{(C,m)*}^{*\prime}(0)} \tag{5.15}$$

En concreto las tasas λ_m y μ_m vendrán dadas por una fracción de $-r_m^{*\prime}(0)$, esto es:

$$\lambda_m = -\frac{\nu_{(C-1,m)} f_{(C-1,m)(C,m)}^{*}(0) + \nu_{(C,m)} f_{(C,m)(C,m)}^{*}(0)}{r_m^{*\prime}(0)}$$
$$\mu_m = -\frac{\nu_{(C-1,m)} f_{(C-1,m)(C-1,m)}^{*}(0) + \nu_{(C,m)} f_{(C,m)(C-1,m)}^{*}(0)}{r_m^{*\prime}(0)}$$

o equivalente a

$$\lambda_m = -\frac{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} f_{(C-1,m)(C,m)}^{*}(0) + f_{(C,m)(C,m)}^{*}(0)}{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} f_{(C-1,m)*}^{*\prime}(0) + f_{(C,m)*}^{*\prime}(0)}$$
$$\mu_m = -\frac{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} f_{(C-1,m)(C-1,m)}^{*}(0) + f_{(C,m)(C-1,m)}^{*}(0)}{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} f_{(C-1,m)*}^{*\prime}(0) + f_{(C,m)*}^{*\prime}(0)} \tag{5.16}$$

Tras simples operaciones algebraicas, la ecuación (5.16) puede expresarse como

$$\lambda_m = \lambda \frac{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} \lambda + [\lambda + m\mu_r + (C-1)\mu)]}{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} (2\lambda + C\mu) + [\lambda + m\mu_r + (2C-1)\mu]}$$
$$\mu_m = [m\mu_r + (C-1)\mu] \frac{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} (\lambda + C\mu) + C\mu}{\frac{\nu_{(C-1,m)}}{\nu_{(C,m)}} (2\lambda + C\mu) + [\lambda + m\mu_r + (2C-1)\mu]} \tag{5.17}$$

Resulta de interés estudiar la relación $\nu_{(C-1,m)} / \nu_{(C,m)}$, la cual aparece de forma explícita en la ecuación (5.17). Tras simples operaciones algebraicas, podemos escribir:

$$
\begin{aligned}
&\frac{v_{(C-1,m)}}{v_{(C,m)}} = \frac{\pi_{2,1}}{\pi_{1,2}} = \\
&= \frac{f^*_{(C,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C-1,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}}}{f^*_{(C-1,m)(C-1,m)}(0)+f^*_{(C-1,m)(C,m)}(0)\frac{p_{(C-1,m+1)(C,m)}}{p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}}} = \\
&= \frac{f^*_{(C,m)(C,m)}(0)p_{(C-1,m+1)(C-1,m)}}{f^*_{(C-1,m)(C-1,m)}(0)[p_{(C-1,m+1)(C-1,m)}+p_{(C-1,m+1)(C,m)}]+f^*_{(C-1,m)(C,m)}(0)p_{(C-1,m+1)(C,m)}} = \\
&= \frac{\lambda(C-1)\mu[\lambda+m\mu_r+(C-1)\mu]}{(\lambda+C\mu)[m\mu_r+(C-1)\mu][(m+1)\mu_r+(C-1)\mu]+\lambda^2(m+1)\mu_r}
\end{aligned}
\tag{5.18}
$$

Es de interés conocer el comportamiento asintótico de λ_m y μ_m y por lo tanto de la relación $v_{(C-1,m)} / v_{(C,m)}$. En esta última se observa que:

$$\text{para } m \to \infty \quad \frac{v_{(C-1,m)}}{v_{(C,m)}} \to \frac{\lambda(C-1)\mu}{(\lambda+C\mu)\mu_r m} \to 0$$

comportamiento que insertado en la expresión (5.17)

$$\text{para } m \to \infty \quad \begin{cases} \lambda_m \to \lambda \\ \mu_m \to C\mu \end{cases}$$

Así, el comportamiento asintótico de este modelo coincide con el modelo propuesto por Falin en [Fal83].

Con esto quedan definidos los parámetros de la aproximación y es posible resolver el sistema de forma muy similar a como se ha resuelto el modelo de Falin. Así, primero se obtienen las probabilidades de estado $\pi(k,m)$ con $0 \leq k \leq C$ y $0 \leq m \leq Q$ que serán las mismas que obtuvimos para el subespacio S_1 del modelo de Falin. A continuación, partiendo de $\pi(C,Q)$, se calculan las probabilidades de estado del subespacio agrupado como:

$$\pi(C,m+1) = \frac{\lambda_m}{\mu_{m+1}}\pi(C,m) \qquad m \geq Q \tag{5.19}$$

Nótese que para calcular $\pi(C,Q+1)$ se puede hacer uso de la misma ecuación de balance, con la salvedad de que la tasa de llegadas a este estado

todavía es λ. Obtenidas las probabilidades de estado, es posible obtener cualquiera de los parámetros de prestaciones que se han comentado a lo largo de este trabajo. En este caso nos centraremos en la probabilidad de bloqueo, que puede se define como:

$$P_b = \sum_{m=0}^{\infty} \pi(C,m) = \sum_{m=0}^{Q-1} \pi(C,m) + \sum_{m=Q}^{\infty} \pi(C,m). \tag{5.20}$$

Si tenemos en cuenta que el comportamiento asintótico del modelo LM, cuando el nivel tiende a infinito, es el mismo que el del modelo [Fal83], podemos simplificar la resolución del sistema tomando la aproximación heterogénea del modelo LM entre los niveles Q y T, con $T \geq Q$ y la aproximación homogénea de Falin a partir del nivel T. De este modo, la suma infinita que aparece en la expresión de la probabilidad de bloqueo se simplifica al aparecer una progresión geométrica a partir de $m = T$.

5.3 HM: modelos de homogeneización

Los modelos propuestos están basados en el modelo de Neuts y Rao definido en [NR90]. En dicho modelo, cuando el número de usuarios en la órbita de reintentos es mayor que Q, la tasa de reintentos toma un valor fijo e igual a $Q\mu_r$. En este apartado se presentan como contribución de la tesis dos nuevos modelos que, partiendo de la idea de que la tasa de reintentos debe ser fija a partir de un determinado nivel —con el fin de homogeneizar el espacio de estados y poder resolver el sistema resultante—, consideran que ésta debe aproximarse lo mejor posible a la tasa de reintentos que en realidad hay en esos estados. De este modo, los modelos propuestos plantean una solución similar a la propuesta en el modelo de Marsan et al [MCL+01] —sección 4.2.2—- y en el modelo FM —sección 4.2.3—. Así, se calcula el número medio de usuarios reintentando, es decir, en la órbita de reintentos, cuando existen Q o más usuarios en la misma, M. De este modo, la tasa de reintentos cuando el número de usuarios en la órbita es mayor que Q pasa de ser $Q\mu_r$ a

$M\mu_r$:

$$\mu_r(m) = \begin{cases} m\mu_r & \text{si } m < Q \\ M\mu_r & \text{si } m \geq Q \end{cases}$$

donde M se define como $M = E[m|m \geq Q]$. Este parámetro se puede calcular del siguiente modo:

$$M = E[m|m \geq Q] = \frac{\sum_{r \geq Q} r\pi_r \mathbf{e}}{\sum_{r \geq Q} \pi_r \mathbf{e}} =$$
$$= \frac{\sum_{u \geq 0} (u+Q)\pi_Q \mathbf{R}^u \mathbf{e}}{\sum_{u \geq 0} \pi_Q \mathbf{R}^u \mathbf{e}} = \frac{\pi_Q \big[\sum_{u \geq 0} u\mathbf{R}^u + Q \sum_{u \geq 0} \mathbf{R}^u \big] \mathbf{e}}{\pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}} =$$
$$= \frac{\pi_Q \big[\mathbf{R}(\mathbf{I} - \mathbf{R})^{-2} + Q(\mathbf{I} - \mathbf{R})^{-1} \big] \mathbf{e}}{\pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}} =$$
$$= \frac{\pi_Q [\mathbf{R}(\mathbf{I} - \mathbf{R})^{-1} + Q\mathbf{I}](\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}}{\pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}} \qquad (5.21)$$

donde $\mathbf{R}$ es la *rate matrix*, $\mathbf{I}$ es la matriz identidad y el vector $\mathbf{e}$ es un vector que tiene todos sus elementos iguales a la unidad. Asimismo, π_m define el vector de probabilidades de estado del nivel m, por ejemplo, $\pi_Q = [\pi(0,Q), \pi(1,Q), \ldots, \pi(C,Q)]$.

Los modelos propuestos mantienen la idea de la homogeneización pero se considera una tasa de reintentos más ajustada a lo que realmente ocurre en el sistema. Con este proceso, al igual que ocurría en el modelo de Neuts y Rao [NR90], el espacio de estados infinito y no homogéneo se aproxima por otro infinito pero homogéneo a partir de un determinado nivel, tal y como se muestra en la figura 5.7. El modelo aproximado resultante se puede resolver haciendo uso de la metodología de Neuts [Neu81]. Así, se define el generador

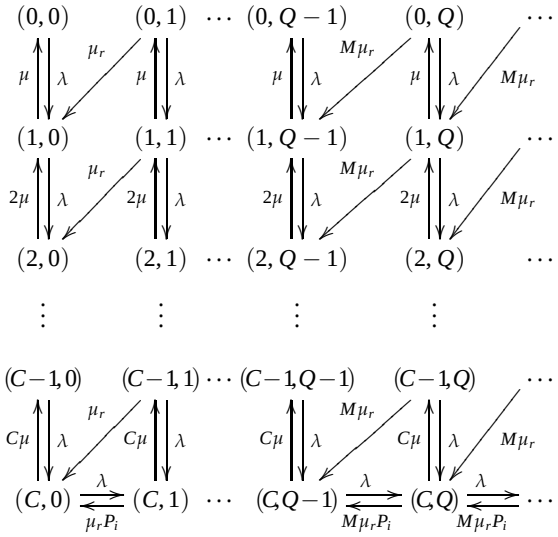


Figura 5.7: Diagrama de transiciones de los modelos HM.

infinitesimal como:

$$\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{A}_1^{(0)} & \mathbf{A}_0^{(0)} & 0 & \dots & 0 & 0 \\ \mathbf{A}_2^{(1)} & \mathbf{A}_1^{(1)} & \mathbf{A}_0^{(1)} & \dots & 0 & \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A}_1^{(Q-1)} & \mathbf{A}_0^{(Q-1)} \\ 0 & 0 & 0 & \dots & \mathbf{A}_2 & \mathbf{A}_1 + \mathbf{R}\mathbf{A}_2 \end{bmatrix},$$

donde las submatrices $\mathbf{A}$ son matrices cuadradas de dimensiones $(C+1) \times (C+1)$ de la forma:

$$\mathbf{A}_{\mathbf{0}}^{(\mathbf{m})} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \lambda \end{bmatrix},$$

$$\mathbf{A_1^{(m)}} = \begin{bmatrix} * & \lambda & 0 & \dots & 0 \\ \mu & * & \lambda & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2^{(m)}} = \begin{bmatrix} 0 & m\mu_r & 0 & \dots & 0 \\ 0 & 0 & m\mu_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & m\mu_r \\ 0 & 0 & 0 & \dots & m\mu_r P_i \end{bmatrix}.$$

Los asteriscos que aparecen en $\mathbf{A_1^{(m)}}$ son valores negativos que hacen que la suma de los elementos de una fila del generador $\mathbf{Q}$ sean cero. Nótese que las submatrices de la última fila del generador infinitesimal son fijas y se corresponden con las matrices de los niveles Q y posteriores.

$$\mathbf{A_1} = \begin{bmatrix} * & \lambda & 0 & \dots & 0 \\ \mu & * & \lambda & \dots & 0 \\ 0 & 2\mu & * & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda \\ 0 & 0 & 0 & \dots & * \end{bmatrix},$$

$$\mathbf{A_2} = \begin{bmatrix} 0 & M\mu_r & 0 & \dots & 0 \\ 0 & 0 & M\mu_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & M\mu_r \\ 0 & 0 & 0 & \dots & M\mu_r P_i \end{bmatrix}.$$

Para resolver este sistema es necesario conocer el valor de M. Este parámetro depende del valor de Q, que es un parámetro de configuración, pero también de las probabilidades de estado en régimen permanente. Aparece, de este modo, un sistema de ecuaciones no lineales que relacionan M, $\mathbf{R}$ y el vector de probabilidades de estado. Para resolver este sistema es necesario hacer uso de un proceso iterativo, que consta de los siguientes pasos:

1. Inicialización: $M(0) = Q$
2. Cálculo de la *rate matrix* $\mathbf{R}$ y las probabilidades de estado para dicho valor de M. Para ello se utiliza alguno de los algoritmos existentes para la obtención de $\mathbf{R}$. Una vez calculada esta matriz, las probabilidades de estado para $m \leq Q$ se pueden obtener como $[\pi_0 \dots \pi_Q]\hat{\mathbf{Q}} = \mathbf{0}$ —junto con la condición de normalización—. Mientras que las probabilidades de estado de aquellos estados de los niveles superiores a Q se calcularán como $\pi_{Q+n} = \mathbf{R}^n \pi_Q$.
3. Cálculo de $M(n+1)$ mediante la ecuación (5.21).
4. Si $|M(n+1) - M(n)| / M(n) \geq \varepsilon$ volver al paso (2).

Al finalizar este proceso iterativo tendremos las probabilidades de estado en régimen permanente que permitirán el cálculo de los parámetros de mérito. En concreto, para la probabilidad de bloqueo tenemos:

$$P_b = \sum_{m=0}^{Q-1} \pi_m \mathbf{z} + \pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{z} \quad con \quad \mathbf{z} = [0, 0, \dots, 0, 1].$$

Mientras que para la probabilidad de servicio inmediato, P_{si}, servicio demorado, P_{sd}, y de no servicio, P_{ns} aparecen las expresiones:

$$P_{si} = 1 - P_b.$$

$$P_{sd} = \lambda^{-1} \mu_r \Big[\sum_{m=0}^{Q-1} m \pi_m \mathbf{o} + M \pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{o} \Big] \quad con \quad \mathbf{o} = [1, 1, \dots, 1, 0].$$

$$P_{ns} = \lambda^{-1} P_i \mu_r \Big[\sum_{m=0}^{Q-1} m \pi_m \mathbf{z} + M \pi_Q (\mathbf{I} - \mathbf{R})^{-1} \mathbf{z} \Big] \quad con \quad \mathbf{z} = [0, 0, \dots, 0, 1].$$

Por último la expresión del número medio de usuarios reintentando viene dada por:

$$N_{ret} = \sum_{m=0}^{Q-1} m\pi_m \mathbf{e} + \pi_Q\big(\mathbf{R}(\mathbf{I}-\mathbf{R})^{-1} + Q\mathbf{I}\big)(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e} \; con \quad \mathbf{e} = [1,1,\ldots,1].$$

5.3.1 Modelo HM1

El modelo HM1 hace referencia a la posibilidad de realizar una aproximación en el cálculo del parámetro M. Se asume que, cuando el número de usuarios en la órbita es suficientemente elevado, es muy probable que todos los servidores estén ocupados. Por tanto, los estados en que existen servidores libres se pueden despreciar. Esta suposición sólo será aceptable en aquellas condiciones que favorecen las transiciones hacia estados con un mayor número de servidores ocupados. En general, estas condiciones se pueden resumir en que la aproximación HM1 será buena cuando la probabilidad de bloqueo sea alta.

Si expresamos esta suposición de forma matemática, tenemos que:

$$\pi_Q \approx \pi(C,Q)\boldsymbol{\psi},$$

donde $\boldsymbol{\psi} = [0 \;\; 0 \;\; \ldots \;\; 0 \;\; 1]^t$. Con esta simplificación se consigue que el parámetro M y la matriz $\mathbf{R}$ dejen de ser dependientes del vector de probabilidades de estado, π_Q. De este modo se consigue simplificar la ecuación para el cálculo de M que, en este caso, queda:

$$M \approx \frac{\boldsymbol{\psi}[\mathbf{R}(\mathbf{I}-\mathbf{R})^{-1} + Q\mathbf{I}](\mathbf{I}-\mathbf{R})^{-1}\mathbf{e}}{\boldsymbol{\psi}(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e}} \tag{5.22}$$

Aunque con este modelo se ha eliminado una de las dependencias existentes entre los diferentes parámetros implicados en la resolución del sistema, la interrelación entre $\mathbf{R}$ y M se mantiene, y por tanto es necesario hacer uso de un proceso iterativo, similar al anterior:

1) inicializamos el proceso con $M(0) = Q$;

2) partiendo del valor de $M(n)$ se calcula la matriz **R** utilizando, por ejemplo, el algoritmo presentado en [LR99, Section 8.4], para calcular posteriormente el valor de $M(n+1)$ usando (5.22);

3) si $|M(n+1) - M(n)| / M(n) \geq \varepsilon$, no se asegura la precisión mínima requerida y es necesario volver al paso (2);

4) Asegurada la tolerancia de M y con la matriz **R** se calculan las probabilidades de estado resolviendo el *QBD* asociado.

5.3.2 Modelo HM2

El modelo HM2, a diferencia de lo que ocurre en el modelo HM1, no introduce ninguna aproximación en el cálculo de M, haciendo uso de la ecuación (5.21) para la obtención de dicho de parámetro. De este modo, el modelo HM2 mejorará los resultados de HM1 consiguiendo buenas precisiones para los diferentes parámetros de mérito, tanto para situaciones con probabilidades de bloqueo altas como bajas. Sin embargo, esta mejora en la precisión del proceso supondrá un mayor coste computacional. Mientras que el modelo HM1 calcula las probabilidades de estado una sola vez, en el modelo HM2 es necesario calcularlas en cada uno de los pasos del proceso iterativo definido para el cálculo de M, aumentando de este modo el coste computacional.

5.4 Análisis comparativo

En esta sección se comparan los resultados de los modelos de truncación generalizada desarrollados. Para evaluar el comportamiento de los diferentes modelos se comparan los resultados en cuanto a complejidad —valor de Q necesario para asegurar una determinada precisión— y tiempo de resolución.

Con el fin de llevar a cabo esta evaluación se ha tomado un conjunto bastante amplio de escenarios, variando tanto la carga del sistema como la tasa de reintentos. Así, partiendo de la carga del sistema definida

Tabla 5.1: Q necesario para garantizar un $\epsilon < 10^{-4}$ en la P_b calculada.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.5	LM	14	13	11	8
	HM1	7	9	**7**	**5**
	HM2	**1**	**5**	**7**	**5**
0.7	LM	34	27	20	14
	HM1	17	17	13	**8**
	HM2	**9**	**6**	**12**	**8**
0.9	LM	172	91	60	38
	HM1	92	52	**33**	**20**
	HM2	**77**	**50**	**33**	**20**

como $\rho = \lambda/(C\mu)$, se ha tomado $C = 50$ y $\mu = 1/180$, variando λ con el fin de evaluar el sistema en diferentes situaciones de carga. Asimismo se han tomado diferentes valores para la tasa de reintentos, en concreto $\mu_r = \{0.001, 0.01, 0.1, 1\}$, variando de este modo el comportamiento de la órbita de reintentos. Nótese además que se ha tomado $P_i = 0$ puesto que, aunque los modelos HM1 y HM2 permiten la existencia de usuarios impacientes, esto no ocurre para el modelo LM.

Se ha centrado el estudio en el cálculo de la probabilidad de bloqueo, así la tabla 5.1 muestra el valor de Q para garantizar un error relativo en esta probabilidad inferior a 10^{-4}. Los modelos HM1 y sobretodo HM2 consiguen mejores resultados que el modelo LM en cualquiera de los escenarios estudiados. Así, estos modelos alcanzan la precisión deseada con un valor de Q menor que el requerido por el modelo LM. Si nos fijamos únicamente en los modelos HM1 y HM2, para los casos con valores altos de μ_r ambos modelos se igualan, sin embargo en el resto de casos el modelo HM2 obtiene mejores resultados que HM1.

Por otro lado se ha calculado el tiempo de cómputo necesario para obtener dicha precisión. Tal y como se observa en la tabla 5.2, podemos concluir que, aunque el valor de Q requerido por el modelo LM es mayor, permite resolver

Tabla 5.2: tiempo (s) para garantizar un $\epsilon < 10^{-4}$ en la P_b.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.5	LM	0.0433	0.0396	0.0327	0.0240
	HM1	0.0869	0.0931	0.0860	0.0505
	HM2	0.0286	0.0475	0.0631	0.0541
0.7	LM	0.1184	0.0868	0.0605	0.0421
	HM1	0.1747	0.1749	0.1561	0.0835
	HM2	0.1391	0.0995	0.1089	0.0899
0.9	LM	1.3427	0.4737	0.2576	0.1387
	HM1	0.8639	0.5476	0.4194	0.2238
	HM2	1.1574	0.5972	0.3079	0.2356

el sistema en un tiempo inferior al requerido por los modelos HM1 y HM2. Este resultado es comprensible si tenemos en cuenta la necesidad de hacer uso de un proceso iterativo para el cálculo del parámetro M en los modelos HM1 y HM2 que, a su vez incluyen otro proceso iterativo para el cálculo de la matriz $\mathbf{R}$. Estos procesos van a ralentizar el proceso de resolución, pero además incluyen cierta varianza en el tiempo de cálculo puesto que dependen del valor inicial escogido. Por otro lado, cabe destacar que los valores que aparecen en la tabla 5.2 corresponden con el tiempo de resolución para el sistema con el valor de Q que garantiza la precisión objetivo en la probabilidad de bloqueo, pero no el proceso de búsqueda de la Q adecuada. Así, si consideramos que dicha búsqueda comenzará con $Q = 1$, es lógico pensar que la diferencia de coste temporal entre unos modelos y otros disminuirá. Esta característica se deriva del hecho de que el modelo LM, que es el más rápido es el que requiere un valor de Q mayor, mientras que los modelos HM1 y HM2 que requieren valores de Q menores son algo más lentos.

Para el caso que nos ocupa, en que el principal objetivo es comparar la eficiencia de diferentes modelos de reintentos, esta diferencia de costes temporales no es significativa, ya que es despreciable desde el punto de vista humano. Sin embargo, dentro del diseño de una red de comunicaciones a nivel global, el tiempo de cómputo puede ser clave a la hora de elegir un

algoritmo u otro.

5.4.1 Conclusiones

Se concluye, por tanto, que para la resolución de sistemas de reintentos, los modelos HM1 y HM2 aseguran que el tamaño del sistema a resolver para garantizar una precisión concreta, es menor que el que necesitaríamos si utilizáramos el modelo LM; haciendo, asimismo, un menor uso de espacio de memoria que el modelo LM. Sin embargo, en términos de coste temporal, el modelo LM supera a los modelos HM1 y HM2.

La literatura especializada, en general, a la hora de comparar los diferentes modelos propuestos se ha centrado principalmente en el punto de truncación, es decir en el valor de Q. Además el modelo LM no permite la existencia de impaciencia en el sistema, mientras que los modelos HM1 y HM2 sí que pueden considerar la existencia de impaciencia. Es por ello que en el siguiente capítulo se utilizarán únicamente los modelos HM1 y HM2 para comparar las prestaciones de los diferentes modelos propuestos frente a los modelos existentes en la literatura.

Capítulo 6

Evaluación de prestaciones

En este capítulo se han evaluado las prestaciones de los modelos propuestos con las soluciones más importantes que podemos encontrar en la literatura. En concreto se evaluarán estos modelos en dos escenarios: el primero de ellos considera la existencia de impaciencia, es decir, permite que un usuario pueda abandonar el sistema sin haber obtenido servicio. Posteriormente se comparan estos modelos en un escenario sin impaciencia con el fin de poder comparar los modelos propuestos con los principales modelos existentes en la literatura.

6.1 Escenario con impaciencia, $P_i \neq 0$

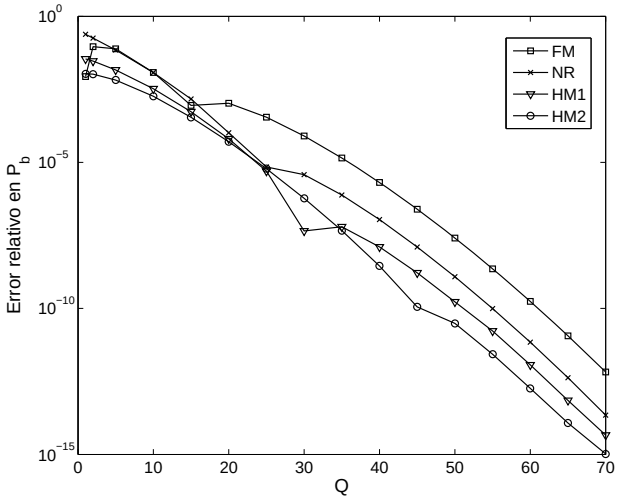
Para realizar la evaluación de prestaciones se ha definido un escenario como el que se mostraba en la figura 5.4 para el cual se definen los siguientes valores $C = 50$ y $\mu = 1/180$. Se ha variado el valor de λ de forma que se evalúe el sistema para diferentes valores de carga, ya que $\rho = \lambda/(C\mu)$. Por otro lado, el valor elegido para la probabilidad de abandonar el sistema tras un reintento fallido es $P_i = 0.2$. Nótese que la introducción del fenómeno de la impaciencia permite considerar valores de carga $\rho > 1$. Asimismo, se

ha considerado oportuno realizar también un estudio en profundidad para diferentes tasas de reintentos, μ_r. De este modo se consigue, sin necesidad de modificar P_i, estudiar el comportamiento de los modelos propuestos con diferentes ocupaciones de la órbita de reintentos para una misma carga del sistema.

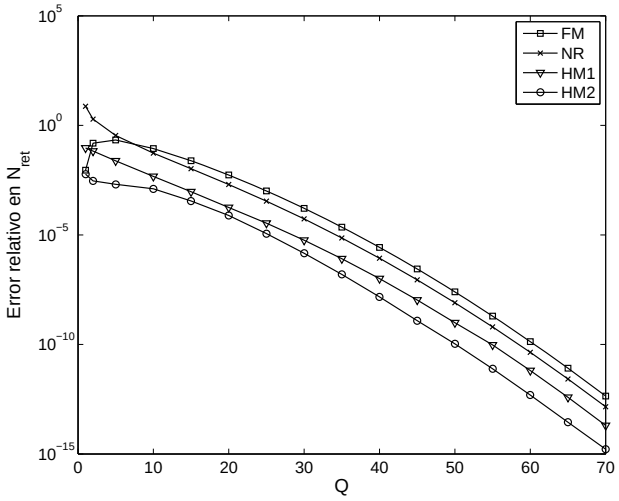
Tomar una P_i diferente de cero evita la comparación con determinados modelos en que no existe esta posibilidad. Así, esta primera comparación sólo considera cuatro modelos: FM, presentado en la sección 5.3; el modelo propuesto por Neuts y Rao en [NR90] —sección 5.1.3, al que denominaremos NR de ahora en adelante—; y los modelos HM1 y HM2 presentados en la sección 5.3. De este modo comparamos nuestros modelos con el modelo [NR90] que puede entenderse como un caso particular de los modelos HM1 y HM2, con el fin de observar las ventajas e inconvenientes de las propuestas realizadas. Pero también el modelo FM propuesto y que presenta un espacio de estados finito, con el fin de comparar estas dos formas de afrontar la resolución de sistemas de reintentos.

La comparación de los diferentes modelos se ha realizando observado el error relativo que se produce en la probabilidad de bloqueo, P_b, en la probabilidad de servicio demorado, P_{sd}, y en la probabilidad de no servicio, P_{ns}, así como el número medio de usuarios reintentando, N_{ret}. La probabilidad de servicio inmediato, P_{si}, al ser la probabilidad complementaría a la probabilidad de bloqueo, puede calcularse a partir de esta última.

Sin embargo, para poder calcular el error relativo es necesario conocer el valor exacto del parámetro de mérito evaluado pero, a diferencia de lo que ocurría en el escenario finito, este valor no es conocido. En este caso es necesario recurrir a una estimación del valor exacto de los parámetros de mérito. Para ello se han ejecutado los diferentes modelos de resolución de sistemas de reintentos aumentando el valor de Q hasta que la diferencia entre dos realizaciones consecutivas —es decir, el valor del parámetro de mérito analizado para dos valores de Q consecutivos— fuese menor que 10^{-14}. Aunque el valor de Q que asegura dicha precisión variará para cada uno de los pa-



(a) P_b.



(b) N_{ret}.

Figura 6.1: Evolución del error relativo con Q.

rámetros de mérito, se ha optado por elegir aquel valor de Q que asegura la estabilidad computacional de P_b y N_{ret} hasta el catorceavo decimal, como valor óptimo para el cálculo del valor exacto de todos los parámetros de mérito. La figura 6.1 muestra el comportamiento del error relativo para los diferentes modelos estudiados conforme aumenta el valor de Q para el caso particular en que $\rho = 0.8$ y $\mu_r = 0.01$. En dicha figura se muestra como el error relativo disminuye conforme aumenta el valor de Q dado que nos acercamos más al modelo exacto. Es importante destacar que el valor exacto calculado por los diferentes modelos es el mismo, lo que cambiará de un modelo a otro será el valor de Q necesario para alcanzar una determinada precisión. Comentar asimismo que, para un mismo valor de Q el modelo HM2 es el que consigue mejores resultados tanto para la probabilidad de bloqueo como para en número medio de usuarios reintentando. Por otro lado, el modelo FM es el que presenta los peores resultados para ambos parámetros.

La Tabla 6.1 muestra los valores de P_b y N_{ret} calculados según el método descrito y que se consideran los valores exactos. Se han estudiado una gran variedad de escenarios en que se ha variado tanto la carga del sistema, ρ, como la tasa de reintentos μ_r. Para un valor fijo de ρ, una disminución en el valor de μ_r supone un incremento de la probabilidad de bloqueo y del número medio de usuarios en la órbita. Obviamente, el aumento de ρ, para un valor fijo de μ_r supondrá un incremento en la probabilidad de bloqueo experimentada por el sistema.

En general, conforme aumentamos ρ y disminuimos μ_r el valor de Q necesario para estabilizar la P_b y N_{ret} aumentan y la complejidad computacional se convierte en un factor decisivo. En el caso que nos ocupa, se ha decidido limitar el estudio a valores de $\rho \leq 10$, puesto que el cálculo del valor exacto de los parámetros de mérito para valores mayores resulta muy costoso, especialmente para $\mu_r = 0.001$. Nótese además que, para dichas configuraciones del sistema, la probabilidad de bloqueo es prácticamente del 100 %. Situación que, en la mayoría de casos prácticos, se considera no viable. Por esa razón, para la comparación de los diferentes modelos se han limitado los valores de carga utilizados a $\rho = \{0.4, 0.6, 0.8, 1.0, 1.2, 1.4\}$, ya que probabili-

Tabla 6.1: Estimaciones de P_b y N_{ret}.

ρ	$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.4	$P_b = 7.664 \cdot 10^{-9}$	$P_b = 7.951 \cdot 10^{-9}$	$P_b = 9.299 \cdot 10^{-9}$	$P_b = 9.360 \cdot 10^{-9}$
	$N_{ret} = 8.558 \cdot 10^{-7}$	$N_{ret} = 9.276 \cdot 10^{-8}$	$N_{ret} = 1.487 \cdot 10^{-8}$	$N_{ret} = 3.017 \cdot 10^{-9}$
0.6	$P_b = 2.262 \cdot 10^{-4}$	$P_b = 2.566 \cdot 10^{-4}$	$P_b = 3.378 \cdot 10^{-4}$	$P_b = 3.050 \cdot 10^{-4}$
	$N_{ret} = 3.800 \cdot 10^{-2}$	$N_{ret} = 4.627 \cdot 10^{-3}$	$N_{ret} = 8.947 \cdot 10^{-4}$	$N_{ret} = 1.531 \cdot 10^{-4}$
0.8	$P_b = 2.548 \cdot 10^{-2}$	$P_b = 3.318 \cdot 10^{-2}$	$P_b = 3.994 \cdot 10^{-2}$	$P_b = 2.897 \cdot 10^{-2}$
	$N_{ret} = 5.883$	$N_{ret} = 0.876$	$N_{ret} = 0.160$	$N_{ret} = 2.009 \cdot 10^{-2}$
1.0	$P_b = 0.347$	$P_b = 0.333$	$P_b = 0.273$	$P_b = 0.172$
	$N_{ret} = 1.359 \cdot 10^{2}$	$N_{ret} = 14.332$	$N_{ret} = 1.573$	$N_{ret} = 0.154$
1.2	$P_b = 0.6482$	$P_b = 0.6387$	$P_b = 0.5436$	$P_b = 0.3541$
	$N_{ret} = 450.19$	$N_{ret} = 44.74$	$N_{ret} = 4.300$	$N_{ret} = 0.3928$
1.4	$P_b = 0.7563$	$P_b = 0.7524$	$P_b = 0.7028$	$P_b = 0.5032$
	$N_{ret} = 745.68$	$N_{ret} = 74.44$	$N_{ret} = 7.290$	$N_{ret} = 0.6700$
2.0	$P_b = 0.8713$	$P_b = 0.8706$	$P_b = 0.8619$	$P_b = 0.7527$
	$N_{ret} = 1598.46$	$N_{ret} = 159.81$	$N_{ret} = 15.935$	$N_{ret} = 1.538$
5.0	$P_b = 0.9613$	$P_b = 0.9612$	$P_b = 0.9607$	$P_b = 0.9539$
	$N_{ret} = 5780.45$	$N_{ret} = 578.04$	$N_{ret} = 57.80$	$N_{ret} = 5.770$
10.0	$P_b = 0.9821$	$P_b = 0.9821$	$P_b = 0.9820$	$P_b = 0.9809$
	$N_{ret} = 12728.44$	$N_{ret} = 1272.84$	$N_{ret} = 127.28$	$N_{ret} = 12.725$

dades de bloqueo superiores a valores entre el 50 % y 75 % —dependiendo del valor de μ_r— son inaceptables. En cuanto a los valores de μ_r elegidos, $\mu_r = \{0.001, 0.01, 0.1, 1\}$, nótese que si $\mu = 1/180$, un valor de $\mu_r = 1$ supone que tenemos, en media, 180 reintentos durante el tiempo de sesión, mientras que un valor $\mu_r = 0.001$ supone sólo una media de 0.18 reintentos por sesión. Con lo que se tienen en cuenta amplio rango de situaciones.

Una vez calculados los valores exactos de los diferentes parámetros de mérito podemos comparar los diferentes modelos, para ello se calcula el valor mínimo de Q que garantiza un determinado error relativo. En la tabla 6.2 se muestra los valores mínimos de Q para garantizar un error relativo en la probabilidad de bloqueo y en el número medio de usuarios reintentando

Tabla 6.2: Q mínima para obtener $\epsilon \leq 10^{-4}$ en P_b y N_{ret}.

		$\mu_r = 0.001$		$\mu_r = 0.01$		$\mu_r = 0.1$		$\mu_r = 1.0$	
		P_b	N_{ret}	P_b	N_{ret}	P_b	N_{ret}	P_b	N_{ret}
$\rho = 0.4$	FM	6	8	8	10	5	8	4	4
	NR	5	9	7	9	**4**	7	4	5
	HM1	5	9	7	8	5	6	**3**	**3**
	HM2	**1**	**2**	**4**	**5**	**4**	**5**	**3**	**3**
$\rho = 0.6$	FM	17	18	15	18	10	11	5	5
	NR	11	16	13	14	8	10	**4**	5
	HM1	10	15	11	13	**5**	8	**4**	**4**
	HM2	**4**	**4**	**6**	**8**	6	**7**	**4**	**4**
$\rho = 0.8$	FM	56	61	30	32	13	12	6	**4**
	NR	43	49	21	29	12	13	5	6
	HM1	33	39	20	22	10	11	**4**	5
	HM2	**16**	**23**	**19**	**20**	**9**	**9**	**4**	**4**
$\rho = 1.0$	FM	264	250	58	54	17	15	6	5
	NR	226	254	50	56	15	17	6	7
	HM1	**137**	211	41	46	13	13	**5**	5
	HM2	144	**190**	**38**	**35**	**12**	**10**	**5**	**4**
$\rho = 1.2$	FM	566	552	87	83	20	18	7	6
	NR	516	564	76	86	18	20	6	7
	HM1	**453**	**498**	63	73	15	16	**5**	5
	HM2	454	500	**62**	**62**	**14**	**11**	**5**	**4**
$\rho = 1.4$	FM	856	837	115	110	23	21	7	6
	NR	792	861	103	117	20	24	6	8
	HM1	**748**	**792**	84	101	**17**	19	6	6
	HM2	**748**	795	**83**	**91**	**17**	**14**	**5**	**5**

Leyenda:
FM: modelo propuesto en la sección 4.2.3
NR: modelo de Neuts y Rao [NR90], sección 5.1.3
HM1: modelo propuesto en la sección 5.3.1
HM2: modelo propuesto en la sección 5.3.2

menor que 10^{-4}. De igual forma, la tabla 6.3 muestra los valores requeridos para la probabilidad de servicio demorado y la de no servicio.

Obsérvese como, para un modelo concreto, el valor de Q necesario para

Tabla 6.3: Q mínima para obtener $\epsilon \leq 10^{-4}$ en P_{sd} y P_{ns}.

		$\mu_r = 0.001$		$\mu_r = 0.01$		$\mu_r = 0.1$		$\mu_r = 1.0$	
		P_{sd}	P_{ns}	P_{sd}	P_{ns}	P_{sd}	P_{ns}	P_{sd}	P_{ns}
$\rho = 0.4$	FM	6	14	7	13	8	9	5	**4**
	NR	4	11	6	11	6	8	**4**	**4**
	HM1	4	11	9	10	7	8	**4**	5
	HM2	**2**	**6**	**3**	**8**	**4**	**6**	**4**	**4**
$\rho = 0.6$	FM	16	27	13	22	11	12	5	5
	NR	11	20	12	18	9	10	5	5
	HM1	17	20	15	18	10	11	5	6
	HM2	**2**	**13**	**4**	**12**	**5**	**8**	**4**	**4**
$\rho = 0.8$	FM	53	76	32	37	14	14	6	5
	NR	42	58	24	32	12	12	**5**	5
	HM1	50	57	28	32	13	14	6	6
	HM2	**18**	**39**	**19**	**22**	**9**	**10**	**5**	**4**
$\rho = 1.0$	FM	266	266	59	58	17	16	7	5
	NR	230	236	51	50	15	14	6	5
	HM1	249	256	54	57	16	17	6	7
	HM2	**182**	**197**	**40**	**40**	**12**	**11**	**5**	**4**
$\rho = 1.2$	FM	571	535	88	77	21	17	7	5
	NR	523	**487**	78	68	18	15	**6**	5
	HM1	560	562	84	86	19	20	7	7
	HM2	**499**	500	**63**	**64**	**15**	**13**	**6**	**3**
$\rho = 1.4$	FM	864	791	118	99	24	18	8	5
	NR	803	**750**	105	**87**	21	16	7	5
	HM1	856	858	115	117	23	24	7	8
	HM2	**794**	795	**92**	91	**18**	**15**	**6**	**4**

Leyenda:
FM: modelo propuesto en la sección 4.2.3
NR: modelo de Neuts y Rao [NR90], sección 5.1.3
HM1: modelo propuesto en la sección 5.3.1
HM2: modelo propuesto en la sección 5.3.2

asegurar la precisión definida cambia con la configuración elegida, es decir con los valores de ρ y μ_r, pero también con el parámetro de mérito analizado. Así pues, a la hora de determinar el valor de Q necesario para asegurar

el correcto funcionamiento del sistema, se debe elegir aquel que asegure la convergencia de todos los parámetros de mérito. En las tablas 6.2 y 6.3, para una configuración y un parámetro de mérito concretos, el valor en negrita indica el mejor modelo de resolución de entre los cuatro comparados. Es decir, el que consigue la precisión objetivo con un valor de Q menor y por tanto, con una complejidad más reducida.

Observando ambas tablas se concluye que el modelo HM2 es el mejor de los cuatro comparados, por ser el que consigue un valor de Q menor para la mayoría de configuraciones y de parámetros de mérito. Destacar en este sentido que, mientras que el modelo HM1 obtiene muy buenos resultados para la P_b y N_{ret} consiguiendo incluso resultados mejores que HM2 en los casos en que $\mu_r = 0.001$ y $\rho \geq 1.0$, para P_{ns} es el modelo NR el que consigue superar HM2 en algunas configuraciones. En todos los casos estudiados el modelo FM es el peor de los cuatro comparados. El hecho de que los modelos HM1, HM2 y NR consigan mejores resultados que el modelo FM nos indica que, en general, los modelos truncados generalizados consiguen una mayor eficiencia que los truncados.

Puesto que estamos trabajando con métodos numéricos es necesario estudiar estos modelos no sólo en términos de la complejidad requerida para conseguir una determinada precisión, sino también en términos del coste computacional. Con el fin de que la evaluación del coste computacional sea lo más equitativa posible se ha utilizado la misma metodología de resolución de *QBDs* en los cuatro modelos. En concreto, se ha utilizado el algoritmo GJL propuesto en [GJL84]. Asimismo, para el proceso iterativo que existe en los tres modelos —en el caso del modelo FM para el cálculo de los parámetros m y p, y en los modelos HM1 y HM2 para el cálculo de M— se ha utilizado una tolerancia de $\epsilon = 10^{-3}$. Adicionalmente, se debe tener en cuenta que el coste temporal dependerá también del software utilizado así como del procesador utilizado. En nuestro caso, todas las pruebas se han realizado con Matlab en un *Intel Pentium Core 2*.

La figura 6.2 muestra el tiempo necesario para resolver los diferentes mo-

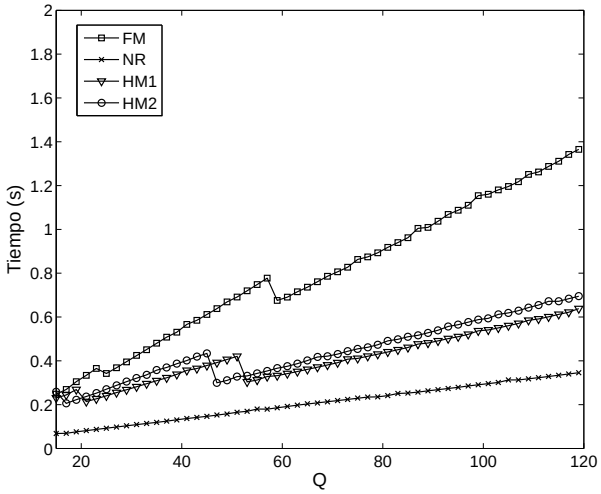


Figura 6.2: Coste computacional para los diferentes modelos desarrollados.

delos conforme varía el valor de Q utilizado, en un escenario con $\rho = 0.8$ y $\mu_r = 10^{-2}$. Como se observa, para un valor de Q el coste temporal de los modelos truncados generalizados es menor que el del modelo FM. De entre todos los modelos truncados generalizados, el más rápido es el NR. Aunque para aquellos casos en que se pretende resolver gran cantidad de sistemas el coste temporal puede ser un factor determinante, desde el punto de vista de la comparación de modelos se puede considerar que el tiempo requerido para resolver estos sistemas con una precisión aceptable —valores de Q alrededor de $30 - 40$ — es despreciable desde un punto de vista humano, no superando el segundo en ninguno de los modelos comparados. Así, a la hora de comparar modelos, la diferencia en coste computacional entre los modelos truncados generalizados es suficientemente pequeña como para no tenerla en cuenta y centrarnos sólo en la complejidad del sistema a resolver, dada en términos del valor de Q necesario, para decidir el modelo a utilizar.

6.2 Escenario sin impaciencia, $P_i = 0$

Aunque el fenómeno de la impaciencia forma parte del comportamiento humano y lo vamos a encontrar en gran parte de las aplicaciones que presentan reintentos, muchos de los modelos definidos para la resolución de sistemas de reintentos no tienen en cuenta esta posibilidad. Es por ello que, si queremos comparar los modelos desarrollados con las soluciones más conocidas en este campo, debemos recurrir a sistemas sin impaciencia. Por lo tanto, a partir de este punto nos centraremos en el caso particular en que $P_i = 0$.

Asimismo, se ha disminuido el número de servidores a $C = 10$ dado que la complejidad computacional del modelo propuesto por Artalejo y Pozo en [AP02] hace imposible la resolución de escenarios con valores de C mayores. Destaca, además, la desaparición de alguno de los parámetros de mérito considerados hasta el momento. Al no existir impaciencia, la probabilidad de no obtener servicio pasa a ser cero. En este mismo sentido, la probabilidad de servicio demorado será equivalente a la probabilidad de bloqueo, $P_{sd} = P_b$, puesto que todos los usuarios que se bloqueen acabarán siendo servidos tras uno o varios reintentos.

Los modelos que podemos encontrar en las secciones 3, 4.1 y 5.1 se pueden clasificar en dos categorías diferentes: aquellos en que el nivel de truncación, determinado por el valor de Q, es un parámetro de configuración del modelo y aquellos que ofrecen una única solución. En el caso de los modelos de la primera categoría, la precisión se ajusta a través de parámetro Q, mientras que en la segunda categoría la precisión es una propiedad intrínseca del modelo y por tanto no se puede ajustar, presentado un valor fijo. Entre los modelos que presentan una precisión ajustable encontramos los modelos de Wilkinson [FT97] —al que denominaremos *Wil* de ahora en adelante—, Falin [Fal83] —*Fal*—, Neuts y Rao [NR90], NR, Artalejo y Pozo [AP02] —*AP*—, así como los modelos FM y HMs propuestos. Entre los modelos en que la precisión es fija tenemos los modelos de Fredericks y Reisner [FR79] —al que denominaremos *FR*—, Greenwerg y Wolff [GW87] —*GW*—, Marsan et al [MCL+01] —*Mar*—, así como los modelos Loss e Interpolación —al que se

Tabla 6.4: Error relativo de P_b para modelos de precisión fija.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$
0.6	FR	$1.399 \cdot 10^{-2}$	$5.462 \cdot 10^{-2}$	$4.603 \cdot 10^{-2}$
	GW	$4.389 \cdot 10^{-2}$	$2.187 \cdot 10^{-1}$	$4.022 \cdot 10^{-1}$
	Loss	$3.468 \cdot 10^{-2}$	$2.112 \cdot 10^{-1}$	$3.964 \cdot 10^{-1}$
	Int	$3.448 \cdot 10^{-2}$	$2.096 \cdot 10^{-1}$	$3.846 \cdot 10^{-1}$
	Mar	$8.604 \cdot 10^{-4}$	$8.39 \cdot 10^{-3}$	$2.066 \cdot 10^{-2}$
	P_b	$5.683 \cdot 10^{-2}$	$6.954 \cdot 10^{-2}$	$9.089 \cdot 10^{-2}$
0.8	FR	$2.811 \cdot 10^{-2}$	$1.022 \cdot 10^{-1}$	$7.913 \cdot 10^{-2}$
	GW	$4.361 \cdot 10^{-2}$	$2.079 \cdot 10^{-1}$	$3.702 \cdot 10^{-1}$
	Loss	$4.172 \cdot 10^{-2}$	$2.064 \cdot 10^{-1}$	$3.690 \cdot 10^{-1}$
	Int	$4.145 \cdot 10^{-2}$	$2.042 \cdot 10^{-1}$	$3.521 \cdot 10^{-1}$
	Mar	$1.117 \cdot 10^{-2}$	$2.325 \cdot 10^{-2}$	$2.398 \cdot 10^{-2}$
	P_b	0.2469	0.2982	0.3750

Leyenda:
FR: modelo de Fredericks y Reisner [FR79] (sección 4.1)
GW: modelo de Greenberg y Wolff [GW87] (sección 3)
Loss: modelo propuesto por Falin [FT97] (sección 3)
Int : modelo propuesto por Falin [FT97] (sección 3)
Mar: modelo de Marsan et al [MCL+01] (sección 4.2.2)

referenciará como *Int*— presentados en [FT97].

Como primera aproximación se estudia la precisión que obtienen los modelos pertenecientes a esta segunda categoría. En la Tabla 6.4 se muestra tanto el valor de la probabilidad de bloqueo, como la precisión que se obtiene con dichos modelos. Nótese que, en este caso, al eliminar la impaciencia ya no es posible considerar valores de ρ superiores a la unidad. Como se muestra en dicha tabla, las precisiones obtenidas son muy bajas, obteniéndose en muchos casos errores relativos inaceptables. De todos los modelos de precisión fija, el que obtiene mejores resultados es el modelo propuesto por Marsan et al en [MCL+01], que es el que presenta errores relativos menores en todos los casos estudiados.

Puesto que los modelos con precisión fija no van a ofrecer una precisión suficiente en la mayoría de casos, centraremos nuestro estudio en los modelos

con precisión configurable. Aunque, con fines comparativos, consideraremos el modelo FM como un límite superior al comportamiento de estos modelos. Recordemos que el modelo FM se puede entender como una generalización y mejora del modelo de Marsan et al que es el modelo de precisión fija que obtiene mejores resultados.

La Tabla 6.5 muestra el valor de Q mínimo para conseguir un error relativo en la probabilidad de bloqueo menor que 10^{-4}. Igual que en casos anteriores, el valor en negrita representa el valor menor y por tanto el mejor modelo para la resolución de dicho escenario. Para cualquier modelo, el aumento de la carga del sistema supone un incremento del valor de Q necesario para conseguir la precisión objetivo. Se observa también como, para una misma carga, valores menores de μ_r requieren una mayor Q para alcanzar la precisión deseada.

Si comparamos los diferentes modelos entre ellos, la primera conclusión que se observa es que los modelos truncados generalizados —Fal, NR, AP, HM1 y HM2— obtienen mejores resultados que los modelos truncados —Wil y FM—. En este caso se han considerado los modelos más importantes que podemos encontrar en la literatura, tanto truncados como truncados generalizados. Por tanto, los resultados obtenidos permiten concluir que los modelos truncados generalizados presentan una mayor eficiencia que los truncados en la resolución de sistemas de reintentos tal y como se sugiere en [FT97]. Si observamos los modelos truncados, Wil y FM, se concluye que FM consigue mejores resultados que Wil para todos los escenarios estudiados. Mientras que si nos centramos en los modelos truncados generalizados, podemos observar como los modelos HM2 y AP son los que presentan mejores resultados. El modelo HM2 obtiene resultados especialmente buenos para valores de μ_r bajos, mientras que el modelo AP presenta los mejores resultados en cuanto a complejidad para valores altos de tasa de reintentos ($\mu_r/\mu > 10$). Nótese también como el modelo HM1 consigue igualar los resultados que obtiene HM2 para valores altos de μ_r, mientras que para valores bajos de μ_r la Q que requiere es mucho mayor que en el caso de usar HM2, pero nunca mayor que la que se requeriría con NR. Esta característica permite concluir

Tabla 6.5: Q mínima para obtener $\epsilon \leq 10^{-4}$ en P_b cuando $P_i = 0$.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.4	Wil	9	10	10	10
	FM	9	9	7	4
	Fal	11	9	7	4
	NR	7	7	5	4
	AP	10	7	**4**	**1**
	HM1	7	6	5	3
	HM2	**3**	**5**	**4**	3
	P_b	$5.633 \cdot 10^{-3}$	$6.442 \cdot 10^{-3}$	$7.998 \cdot 10^{-3}$	$8.696 \cdot 10^{-3}$
0.6	Wil	24	19	18	17
	FM	23	16	11	5
	Fal	25	16	11	7
	NR	18	13	9	6
	AP	22	13	**6**	**1**
	HM1	15	10	7	4
	HM2	**5**	**9**	7	4
	P_b	$5.683 \cdot 10^{-2}$	$6.954 \cdot 10^{-2}$	$9.089 \cdot 10^{-2}$	$9.979 \cdot 10^{-2}$
0.8	Wil	74	45	40	39
	FM	68	34	17	21
	Fal	70	35	23	14
	NR	53	27	17	10
	AP	57	24	**9**	**1**
	HM1	41	21	13	7
	HM2	**36**	**20**	13	7
	P_b	0.2469	0.2982	0.3750	0.4043

Leyenda:
Wil: modelo de Wilkinson [FT97] (sección 4.1)
FM: modelo propuesto en la sección 4.2.3)
Fal: modelo de Falin [Fal83] (sección 5.1.1)
NR : modelo de Neuts y Rao [NR90] (sección 5.1.3)
AP: modelo de Artalejo y Pozo [AP02] (sección 5.1.2)
HM1: modelo propuesto en la sección 5.3.1
HM2: modelo propuesto en la sección 5.3.2

que este modelo es un punto intermedio entre NR y HM2.

A parte del estudio de la complejidad del sistema, interesa estudiar el cos-

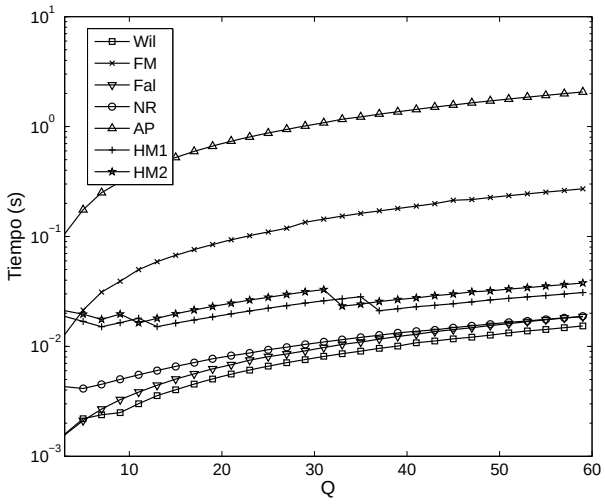


Figura 6.3: Coste computacional para los diferentes modelos cuando $P_i = 0$.

te computacional de dichos modelos. La figura 6.3 muestra el coste temporal asociado a cada uno de los modelos comparados cuando $\rho = 0.8$ y $\mu_r = 0.01$. Los resultados obtenidos muestran que, para un valor de Q dado, los modelos más sencillos como son Wilkinson, Falin o NR consiguen los mejores resultados, por requerir menos operaciones. Detrás de estos modelos podemos encontrar los modelos HM1 y HM2. Finalmente, los modelos con mayor coste computacional son los modelos FM y AP que presentan tiempos de resolución mucho más elevados. Así, aunque en términos de complejidad HM2 y AP eran los mejores modelos, a la hora de resolver interesa utilizar HM2 puesto que el coste computacional es mucho menor.

Sin embargo, también es cierto que para garantizar un óptimo funcionamiento cada modelo requerirá de un valor de Q diferente, por lo que necesitamos considerar el tiempo requerido para resolver el sistema que asegura la precisión objetivo. En la tabla 6.6 se observa el tiempo necesario para resolver el sistema que garantiza un error relativo en la probabilidad de bloqueo

Tabla 6.6: Tiempo de cálculo (s) para obtener $\epsilon \leq 10^{-4}$ en P_b con $P_i = 0$.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.4	Wil	0.0222	0.0187	0.0177	0.0178
	FM	0.0931	0.0999	0.2477	0.1346
	Fal	0.0186	0.0118	0.0111	0.0109
	NR	0.0213	0.0232	0.0168	0.0155
	AP	0.9345	0.3624	0.1292	0.0350
	HM1	0.0227	0.0147	0.0094	**0.0073**
	HM2	**0.0112**	**0.0098**	**0.0090**	0.0079
0.6	Wil	0.0408	0.0326	0.0310	0.0291
	FM	0.2804	0.3466	0.5120	0.2547
	Fal	**0.0234**	**0.0170**	0.0144	0.0118
	NR	0.0502	0.0381	0.0277	0.0216
	AP	2.0150	0.5174	0.1934	0.0392
	HM1	0.0311	0.0246	0.0192	**0.0101**
	HM2	0.0310	0.0237	**0.0143**	0.0105
0.8	Wil	0.1238	0.0761	0.0681	0.0689
	FM	1.6149	2.7071	1.2518	1.5790
	Fal	**0.0540**	**0.0319**	**0.0244**	0.0183
	NR	0.1357	0.0731	0.0511	0.0357
	AP	4.5531	0.7697	0.3228	0.0510
	HM1	0.0804	0.0473	0.0350	**0.0182**
	HM2	0.1062	0.0498	0.0253	0.0185

Leyenda:
Wil: modelo de Wilkinson [FT97] (sección 4.1)
FM: modelo propuesto en la sección 4.2.3)
Fal: modelo de Falin [Fal83] (sección 5.1.1)
NR : modelo de Neuts y Rao [NR90] (sección 5.1.3)
AP: modelo de Artalejo y Pozo [AP02] (sección 5.1.2)
HM1: modelo propuesto en la sección 5.3.1
HM2: modelo propuesto en la sección 5.3.2

menor que 10^{-4}. Como se observa en esta tabla, para las cargas medias / altas con tasas de reintento bajas, un modelo sencillo como el propuesto por Falin consigue muy buenos resultados. En el resto de casos, los modelos con mejores resultados son los modelos HM1 y HM2. De todos modos, cabe destacar que, exceptuando los modelos FM y AP, en resto de modelos consigue resol-

ver cualquiera de los escenarios en un tiempo menor al segundo y por tanto, se puede considerar despreciable desde el punto de vista humano.

Uniendo los resultados de la tabla 6.5 y la figura 6.3 podemos representar el tiempo requerido para conseguir una determinada precisión tal y como se observa en la figura 6.4 para el escenario anterior —$\rho = 0.8$ y $\mu_r = 0.01$—. En dicha figura se observa como los modelos más sencillos, Wil, Fal y NR, obtienen los mejores resultados en cuanto a tiempo, siendo los que consiguen resolver el sistema que garantiza una determinada precisión en menor tiempo. Seguidos de los modelos HM1 y HM2, que destacan, no sólo por el tiempo de computo, sino porque aumentar la precisión supone un incremento despreciable del tiempo de cómputo. Por otro lado, si observamos la precisión, tenemos que los modelos HM1 y HM2 son los que van a permitir obtener errores relativos en la probabilidad de bloqueo menores. Además estos modelos garantizan que, incluso para valores de Q muy bajos –primeros puntos de las curvas— se obtienen errores bajos, inferiores al 5 %. Con otros modelos como Fal y Wil o incluso NR, para valores similares de Q pueden presentar errores del 25 %. Destacar también que los modelos FM y AP no sólo van a ser los más lentos, sino que intentar aumentar la precisión llevará asociado un incremento en el coste temporal considerable. Por otro lado, presentan la ventaja de ofrecer buenas precisiones en la probabilidad de bloqueo par valores de Q bajos.

6.3 Conclusiones

El estudio de diferentes modelos truncados generalizados para la resolución de sistemas de reintentos nos permite concluir que, en general, estos modelos ofrecen soluciones más eficientes que los modelos truncados puesto que la aproximación realizada en el caso de los modelos truncados generalizados es menos agresiva respecto al modelo exacto. Así el hecho de mantener un espacio de estados infinito va a mejorar la precisión de resolución obtenida.

De entre las soluciones truncadas generalizadas, aquellas basadas en la

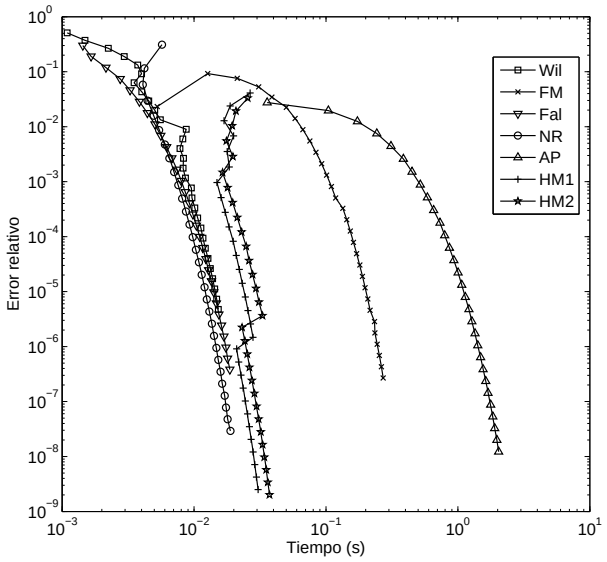


Figura 6.4: Tiempo de ejecución para determinadas precisiones.

homogeneización del espacio de estado consiguen un mejor compromiso entre complejidad y coste computacional que las basadas en la limitación del espacio de estados. Estas últimas se caracterizan por conseguir muy buenos resultados en cuanto a tiempo, o en cuanto a precisión, a costa de empeorar considerablemente el parámetro complementario. Así, mientras que el algoritmo de Falin consigue muy buenos resultados en tiempo gracias a su simplicidad, necesita una Q muy elevada para conseguir la precisión objetivo. Todo lo contrario ocurre con el algoritmo AP, que requiere una Q muy pequeña, pero a cambio necesita de mucho tiempo de procesado.

Si nos centramos en los modelos truncados generalizados basados en la homogeneización del espacio de estados podemos destacar que, en cualquier caso, para los valores de Q de interés, los tiempos de resolución no superan el segundo y por tanto, desde el punto de vista humano, es indiferente el

uso de cualquiera de ellos. De esta forma el parámetro determinante para elegir uno modelo u otro será la precisión. En este sentido, se observar como los modelos HM1 y HM2, en especial este segundo, consiguen muy buenos resultados en precisión. Así, el modelo HM2 es el mejor de los modelos existentes en la literatura cuando la tasa de reintentos es baja, mientras que para tasas elevadas sólo se ve superado por el modelo AP. Nótese sin embargo que el modelo AP puede presentar problemas de resolución para sistemas grandes, es decir con un número de servidores elevado. Otro factor a favor de los modelos HM1 y HM2 es la posibilidad de aplicarlos a escenarios con impaciencia, cosa que no es cierta para otros modelos con un buen resultado de precisión como es el caso del modelo AP o con tiempos de resolución muy bajos como es el caso del modelo de Falin.

Capítulo 7

Otros modelos

Los modelos aproximados que hemos visto hasta el momento están basados en la resolución numérica de las ecuaciones de estado de un proceso de Markov para la CTMC que describe el sistema estudiado. Esto supone basar el proceso de resolución en la obtención de las probabilidades de estado y, a partir de estas, calcular los diferentes parámetros de mérito de interés.

Recientemente, sin embargo, ha aparecido una aproximación alternativa para evaluar procesos de Markov, incluidos aquellos con un espacio de estados infinito. Esta aproximación se denomina *Value Extrapolation* (VE) y ha sido introducida por Leino y otros en [LPV06, LV06]. Esta metodología ya no se basa en obtener las probabilidades de estado sino que, definiendo el sistema como un *Markov Decision Process*, MDP, permite calcular directamente los parámetros de interés, ya sean las propias probabilidades de estado, o parámetros de mérito como la probabilidad de bloqueo y el número medio de usuarios en la órbita de reintentos.

Hasta el momento esta metodología ha sido utilizada en colas monoservidor aunque multiclase, para las que se han obtenido muy buenos resultados como se muestra en [LPV06]. El objetivo de esta sección es aplicar esta metodología a una cola multiservidor, como la que podemos encontrar en un sistema de reintentos. Pero primero comentaremos los principios básicos de

los MDPs que servirán de base para un mejor entendimiento del modelo VE.

7.1 Markov Decision Process, MDP

7.1.1 MDP discreto y políticas de decisión

En primer lugar introducimos un modelo de decisión de Markov. Consideremos un sistema dinámico cuyo estado se observa en instantes de tiempo equidistantes y numerables $t = 0, 1, 2, 3, ...$ El conjunto de estados se denota por I. En dichos instantes el sistema se revisa tomando una determinada decisión o acción. Para cada estado $i \in I$, se dispone de un conjunto $A(i)$ de acciones o decisiones. El espacio de estados I y el conjunto de acciones $A(i)$ se suponen finitos. Las consecuencias económicas de las decisiones que se toman en los instantes de revisión quedan reflejadas en costes. Este sistema controlado dinámicamente se denomina modelo de decisión de Markov discreto cuando cumple las siguientes propiedades. Si en un instante de decisión, estando en el estado i se escoge la acción a, entonces independientemente de la historia del sistema, tenemos que:

- Hay un coste asociado $c_i(a)$
- En el próximo instante de decisión, el sistema estará en el estado j con probabilidad $p_{ij}(a)$ en donde:

$$\sum_{j \in I} p_{ij}(a) = 1, \quad i \in I$$

Hacemos notar que tanto los costes $\{c_i(a)\}$ como las probabilidades de transición $p_{i,j}(a)$ se suponen homogéneas. El coste "'inmediato"' $c_i(a)$ puede interpretarse como el coste en que se incurre hasta el próximo instante de decisión, cuando se ha elegido la acción a en el estado i.

La regla o política de control del sistema dinámico en principio puede ser bastante complicada en el sentido de que la acción a tomar puede depender

de la historia del sistema. No obstante, en vista de la suposición Markoviana y del hecho de que se planifica a largo plazo, habitualmente sólo se consideran políticas estacionarias. Una política estacionaria α es una regla que siempre escoge una acción única α_i cuando el sistema se encuentra en el estado i en el instante de decisión t.

Definamos para $n = 0, 1, , 2, ...$

$X_n =$ El estado del sistema en el instante de decisión n-ésimo

Bajo la condición de política estacionaria α tenemos que

$$P\{X_{n+1} = j | X_n = i\} = p_{ij}(\alpha_i)$$

con independencia de la historia del sistema hasta el instante n. Así bajo una política estacionaria α el proceso estocástico $\{X_n\}$ es un proceso discreto (cadena) de Markov, cuyas probabilidades de transición entre instantes consecutivos de observación, i y j, resultan ser $p_{ij}(\alpha)$. La cadena de Markov incurre en un coste $c_i(\alpha_i)$ cada vez que el sistema visita el estado i.

Con miras a observar el comportamiento del coste del proceso en estudio a largo término, haremos algunas anotaciones previas. Bajo una política estacionaria α, denotemos las probabilidades de transición entre n instantes consecutivos del proceso $\{X_n\}$ como

$$P\{X_{n+1} = j | X_0 = i\} = p_{ij}^{(n)}(\alpha), \quad i, j \in I \quad \text{y} \quad n = 1, 2, ..$$

donde $p_{ij}^{(1)}(\alpha) = p_{ij}(\alpha_i)$. Nótese que tales probabilidades de transición satisfacen la relación de Chapman-Kolmogorov:

$$p_{ij}^{(n+1)}(\alpha) = \sum_{k \in I} p_{ik}^{(n)}(\alpha) p_{kj}(\alpha_k), \quad n = 1, 2, ..$$

Se dice que el estado j es alcanzable desde el estado i con la política α si $p_{ij}^{(n)}(\alpha) > 0$ para algún valor de $n \geq 1$.

Así para cada política estacionaria α, la distribución de equilibrio (probabilidades en régimen permanente) $\{\pi_j(\alpha)\}, j \in I$, cumplen la expresión:

$$\pi_j(\alpha) = \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} p_{ij}^{(n)}(\alpha) \tag{7.1}$$

El límite de (7.1) existe y es independiente del estado inicial $X_0 = i$. La distribución en equilibrio satisface el siguiente sistema de ecuaciones lineales:

$$\begin{aligned} &\pi_j(\alpha) = \sum_{k\in I} \pi_k(\alpha) p_{kj}(\alpha_k), \quad j \in I. \\ &\sum_{j\in I} \pi_j(\alpha) = 1. \end{aligned} \tag{7.2}$$

Este sistema de ecuaciones lineales tiene una única solución.

Sea $g(\alpha)$ el coste promediado por unidad de tiempo cuando se utiliza la política estacionaria α. Podemos afirmar que

$$g(\alpha) = \sum_{j\in I} \pi_j(\alpha) c_j(\alpha_j) \tag{7.3}$$

7.1.2 Valores relativos asociados a una política dada α

Introduciremos los *relative values* bajo una política estacionaria α dada, considerando los costes asociados hasta el primer retorno a un estado de regeneración dado. Sea r el estado en cuestión. Entonces, para cada estado $i \in I$ definimos:

$T_i(\alpha) =$ Tiempo medio esperado hasta el primer retorno al estado r cuando se comienza en el estado i, con la política estacionaria $\alpha, \forall i \in I$.

En particular, si un ciclo se define como el tiempo transcurrido entre dos visitas consecutivas al estado de regeneración r bajo la política estacionaria α, tendremos que $T_r(\alpha)$ es la longitud esperada de un ciclo. También definimos:

$K_i(\alpha) =$ Coste medio esperado hasta el primer retorno al estado r cuando se comienza en el estado i, con la política estacionaria $\alpha, \forall i \in I$.

Igualmente, para un ciclo definido como el tiempo transcurrido entre dos visitas consecutivas al estado de regeneración r bajo la política estacionaria α, tendremos que $K_r(\alpha)$ es el coste medio esperado por ciclo.

Para cualquier valor de $i \in I$, $T_i(\alpha)$ y $K_i(\alpha)$ pueden expresarse como

$$\begin{aligned} T_i(\alpha) &= 1 + \sum_{\substack{j \in I \\ j \neq r}} p_{ij}(\alpha_i) T_j(\alpha), \quad \forall i \in I. \\ K_i(\alpha) &= c_i(\alpha_i) + \sum_{\substack{j \in I \\ j \neq r}} p_{ij}(\alpha_i) K_j(\alpha), \quad \forall i \in I. \end{aligned} \tag{7.4}$$

en donde el valor unitario a la derecha del primer signo de igualdad es debido a la equidistancia entre instantes consecutivos de observación. El valor de $c_i(\alpha_i)$ corresponde al coste asociado a la primera decisión tomada en el estado inicial i.

Para el caso de $i = r$, aplicando resultados de la teoría de procesos de renovación-recompensa (*renewal - reward processes*), el coste medio por unidad de tiempo iguala el coste esperado incurrido en un ciclo dividido por el tiempo medio esperado de un ciclo, esto es:

$$g(\alpha) = \frac{K_r(\alpha)}{T_r(\alpha)}. \tag{7.5}$$

Definamos ahora los *relative values*, $w_i(\alpha)$, como

$$w_i(\alpha) = K_i(\alpha) - g(\alpha) T_i(\alpha), \quad \forall i \in I. \tag{7.6}$$

En (7.6) se observa que si $i = r$ entonces $w_r(\alpha) = 0$. Insertando las expresiones (7.4) en (7.6) nos dan:

$$\begin{aligned} w_i(\alpha) &= c_i(\alpha_i) - g(\alpha) + \sum_{j\in I} p_{ij}(\alpha_i)\{K_j(\alpha) - g(\alpha)T_j(\alpha)\} = \\ &= c_i(\alpha_i) - g(\alpha) + \sum_{j\in I} p_{ij}(\alpha_i) w_j(\alpha), \quad \forall i \in I. \end{aligned} \tag{7.7}$$

Por otra parte, sea $V_n(i,\alpha)$ el coste total esperado al cabo de los n primeros momentos de decisión contados a partir de un estado inicial i y utilizando la política estacionaria α. Teniendo en cuenta los costes individuales en los instantes de decisión $t = 0, 1, 2, ..., n-1$; $V_n(i,\alpha)$ puede escribirse como

$$\begin{aligned} V_n(i,\alpha) &= \sum_{j\in I} p_{ij}^{(0)}(\alpha)c_j(\alpha_j) + \sum_{j\in I} p_{ij}^{(1)}(\alpha)c_j(\alpha_j) + \cdots + \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)c_j(\alpha_j) = \\ &= \sum_{t=0}^{n-1}\sum_{j\in I} p_{ij}^{(t)}(\alpha)c_j(\alpha_j) = V_{n-1}(i,\alpha) + \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)c_j(\alpha_j) \end{aligned}$$

siendo $\sum_{j\in I} p_{ij}^{(t)}(\alpha)c_j(\alpha_j)$ el coste esperado asociado a la acción o decisión t. También hacemos notar el hecho de que $V_{n-1}(i,\alpha) \leq V_n(i,\alpha)$ para $n = 1, 2, ...$

El sistema de ecuaciones lineales (7.7) tiene como incógnitas $g(\alpha)$ y $\{w_i(\alpha)\}$. Sea $\{g, v_i\}$ una solución al mismo. Se puede probar por inducción que

$$v_i = V_n(i,\alpha) - ng + \sum_{j\in I} p_{ij}^{(n)}(\alpha)v_j, \quad \forall i \in I.$$

Admitamos que el conjunto de valores $\{v_i\}$ permanecerá acotado a largo término, esto es, para valores de n muy grandes. Dividiendo la anterior expresión por n y pasando al límite $n \to \infty$ tenemos que:

$$g(\alpha) = \lim_{n\to\infty} \frac{V_n(i,\alpha)}{n}, \quad \forall i \in I. \tag{7.8}$$

y $g = g(\alpha)$ diviendo (7.8) por n y haciendo $n \to \infty$.

Por otra parte, sean $\{g, v_i\}$ y $\{g', v'_i\}$ dos soluciones a (7.7). Dado que $g = g'$ podemos escribir

$$v_i - v'_i = \sum_{j \in I} p_{ij}^{(n)}(\alpha)(v_j - v'_j), \quad \forall i \in I, n \geq 1. \tag{7.9}$$

Sumando (7.9) para $n = 1, 2, ..., m$ y dividiendo por m resulta:

$$\begin{aligned} v_i - v'_i =& \frac{1}{m} \sum_{n=1}^{m} \sum_{j \in I} p_{ij}^{(n)}(\alpha)(v_j - v'_j) = \\ =& \sum_{j \in I} \{ \frac{1}{m} \sum_{n=1}^{m} p_{ij}^{(n)}(\alpha) \}(v_j - v'_j), \quad \forall i \in I, m \geq 1. \end{aligned}$$

Teniendo en cuenta (7.1) al hacer que $m \to \infty$ en la anterior expresión:

$$v_i - v'_i = \sum_{j \in I} \pi_j(\alpha)(v_j - v'_j), \quad \forall i \in I, n \geq 1. \tag{7.10}$$

En donde observamos que los términos a la derecha del signo de igualdad no dependen de i por lo que la diferencia $v_i - v'_i$ será igual a una constante, esto es, $v_i - v'_i = \xi$. Como consecuencia, para resolver el sistema de ecuaciones dado por (7.7) podemos fijar un *relative value*, $v_s = 0$ para algún $s \in I$ y resolverlo para el resto de valores $\{v_s\}$ y para $g(\alpha)$.

7.1.3 Ecuaciones de Howard para el caso de tiempo continuo

De forma similar al caso de una cadena de Markov, tiempo discreto, igualmente podemos formular las ecuaciones de Howard para un proceso de Markov, tiempo continuo. Para tal fin, recordemos que todo proceso de Markov tiene implícitamente asociada una cadena de Markov cuyas probabilidades

de transición $\{p_{ij}\}$ vienen dadas en función de los elementos del generador infinitesimal, $\mathbf{Q}$, esto es, del conjunto de tasas $\{q_{ij}\}$:

$$p_{ij} = \begin{cases} \frac{q_{ij}}{q_i}, & \text{cuando } i \neq j \\ 0, & \text{cuando } i = j \end{cases} \tag{7.11}$$

siendo $q_i = -q_{ii} = \sum_{j \neq i} q_{ij}$.

En este caso, el modelo de Markov con toma de decisiones o acciones, tendría las siguientes características:

- $p_{ij}(a)$ = Probabilidad de que, tomando la acción a en el estado presente i, en el siguiente instante de decisión el sistema alcance el estado j.
- $\tau_i(a)$ = Tiempo medio entre el instante de la decisión actual a -tomada en el estado presente i- y el instante de la próxima decisión o acción.
- $c_i(a)$ = El coste esperado por unidad de tiempo en que se incurre hasta la próxima decisión, si la acción o decisión a se toma en el estado presente i.

De forma paralela a la sección 7.1.2, tendríamos que $T_i(\alpha)$ y $K_i(\alpha)$ pueden expresarse como

$$\begin{aligned} T_i(\alpha) &= \tau_i(a) + \sum_{\substack{j \in I \\ j \neq r}} p_{ij}(\alpha_i) T_j(\alpha), \quad \forall i \in I. \\ K_i(\alpha) &= c_i(\alpha_i)\tau_i(a) + \sum_{\substack{j \in I \\ j \neq r}} p_{ij}(\alpha_i) K_j(\alpha), \quad \forall i \in I. \end{aligned} \tag{7.12}$$

en donde cabe observar la diferencia entre (7.4) -caso discreto- y (7.12) -caso continuo-; en particular en los términos que inmediatamente siguen a los signos de igualdad. Insertando (7.11) y (7.12) en (7.6) nos dan:

$$w_i(\alpha) = [c_i(\alpha_i) - g(\alpha_i)]\tau_i(\alpha_i) + \sum_{j\in I} \frac{q_{ij}(\alpha_i)}{q_i(\alpha_i)}\{K_j(\alpha) - g(\alpha)T_j(\alpha)\} =$$

$$= \tfrac{c_i(\alpha_i)-g(\alpha)}{q_i(\alpha_i)} + \sum_{j\in I} \frac{q_{ij}(\alpha_i)}{q_i(\alpha_i)} w_j(\alpha), \quad \forall i \in I.$$

o equivalentemente

$$q_i(\alpha_i)w_i(\alpha) = c_i(\alpha_i) - g(\alpha) + \sum_{j\in I} q_{ij}(\alpha_i)w_j(\alpha), \quad \forall i \in I. \tag{7.13}$$

Dado que $q_i = \sum_{j\in I} q_{ij}$ la anterior expresión adopta la siguiente forma alternativa

$$c_i(\alpha_i) - g(\alpha) + \sum_{j\in I} q_{ij}(\alpha_i)[w_j(\alpha) - w_i(\alpha)] = 0, \quad \forall i \in I. \tag{7.14}$$

Finalmente indicar que en el caso continuo, $g(\alpha)$ puede derivarse según sigue (aunque g es una incógnita que se obtiene al resolver el sistema de ecuaciones lineales (7.14), con la condición de $v_s = 0$ para algún $s \in I$). Sea $Z_t(i,\alpha)$ el coste medio total acumulado hasta el instante t, a partir de un estado inicial i y bajo una política estacionaria α.

$$g_i(\alpha) = \lim_{t\to\infty} \frac{Z_t(i,\alpha)}{t}, \quad \forall i \in I.$$

Dicho límite puede obtenerse como indicamos en el siguiente esbozo de demostración. Consideremos las primeras m decisiones o acciones, entonces:

$$\lim_{t\to\infty} \frac{Z_t(i,\alpha)}{t} = \lim_{m\to\infty} \frac{E(\text{Coste acumulado en las primeras } m \text{ decisiones})}{E(\text{tiempo transcurrido en las primeras } m \text{ decisiones})}.$$

Veamos cómo calcular el numerador y denominador de la anterior fracción. Sean C_n el coste asociado entre la decisión o acción $n-1$-ésima y la

decisión o acción n-ésima. Sea τ_n el tiempo transcurrido entre la decisión o acción $n-1$-ésima y la decisión o acción n-ésima. Podemos escribir:

$$E(C_n|X_0 = i) = \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)c_j(\alpha_j)$$
$$E(\tau_n|X_0 = i) = \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)\tau_j(\alpha_j)$$

Por lo tanto tendremos que

$$\lim_{m\to\infty} \tfrac{1}{m} \sum_{n=1}^{m} E(C_n|X_0 = i) = \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)c_j(\alpha_j) =$$
$$= \sum_{j\in I} \Big\{ \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} p_{ij}^{(n-1)}(\alpha)\Big\}c_j(\alpha_j) = \sum_{j\in I} \pi_j(\alpha)c_j(\alpha_j)$$
$$\lim_{m\to\infty} \tfrac{1}{m} \sum_{n=1}^{m} E(\tau_n|X_0 = i) = \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} \sum_{j\in I} p_{ij}^{(n-1)}(\alpha)\tau_j(\alpha_j) =$$
$$= \sum_{j\in I} \Big\{ \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} p_{ij}^{(n-1)}(\alpha)\Big\}\tau_j(\alpha_j) = \sum_{j\in I} \pi_j(\alpha)\tau_j(\alpha_j)$$

en donde las dos últimas igualdades provienen de aplicar el resultado de (7.1). Por consiguiente $g_i(\alpha)$ es independiente del estado inicial i y viene dado por [Tij86]:

$$g(\alpha) = g_i(\alpha) = \lim_{t\to\infty} \frac{Z_t(i,\alpha)}{t} = \frac{\sum_{j\in I} c_j(\alpha_j)\pi_j(\alpha)}{\sum_{j\in I} \tau_j(\alpha_j)\pi_j(\alpha)} \tag{7.15}$$

Llegado a este punto, conviene observar las diferencias entre las expresiones de $g(\alpha)$ del caso continuo (7.15) con la del caso discreto (7.3).

7.2 Value Extrapolation

La aplicación de VE a la resolución de sistemas de reintentos requiere que el espacio de estados a resolver sea finito. Sin embargo, puesto que el escenario

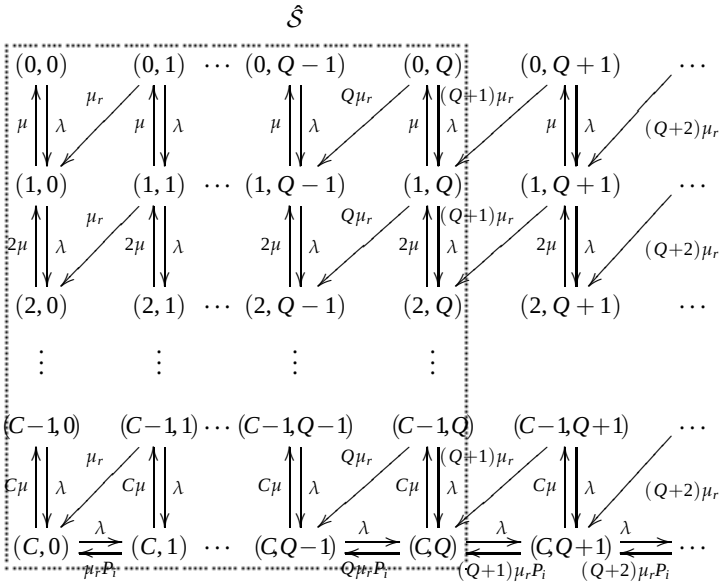


Figura 7.1: Diagrama de transiciones del modelo VE.

considerado es el mismo que hemos utilizado para la comparación de los modelos truncados generalizados, el espacio de estados del sistema resolver será infinito, tal y como se observa en el diagrama de transiciones de la figura 7.1. Así, la primera acción a realizar será truncar el espacio de estados limitando el número máximo de usuarios en la órbita de reintentos, $m \leq Q$. Al espacio truncado lo denominaremos $\hat{\mathcal{S}}$ tal y como se puede ver en la misma figura.

Definido el espacio de estados truncado, $\hat{S}$, el modelo VE debe definir el sistema como un MDP. Para ello se parte de las ecuaciones de Howard que vienen dadas por la expresión:

$$c_s - g + \sum_{s'} q_{ss'}(w_{s'} - w_s) = 0 \quad \forall s,$$

donde c_s es el coste / recompensa de realizar una acción en el estado s, g es el

coste promediado, $q_{ss'}$ tasas de transición de un estado s a otro s' y w_s es el *relative state value*.

Aparecen tantas ecuaciones de Howard como número de estados haya en el problema, $|\hat{S}|$, mientras que el número de incógnitas es de $|\hat{S}|+1$ — los $|\hat{S}|$ *relative state values* más el coste promediado, g—. Sin embargo, como estas ecuaciones definen las diferencias en los *relative state values*, se toma $w_0 = 0$ con el fin de tener el mismo número de ecuaciones que de incógnitas y poder resolver el sistema de ecuaciones.

Para el sistema que nos ocupa, estas ecuaciones, con $m \leq Q$, vienen dadas por:

Para $k < C$:

$$c_{(k,m)} - g + \lambda[w_{(k+1,m)} - w_{(k,m)}] + k\mu[w_{(k-1,m)} - w_{(k,m)}] + \\ + m\mu_{ret}[w_{(k+1,m-1)} - w_{(k,m)}] = 0$$

Para $k = C$:

$$c_{(C,m)} - r + \lambda[w_{(C,m+1)} - w_{(C,m)}] + C\mu[w_{(C-1,m)} - w_{(C,m)}] + \\ + m\mu_{ret}P_i[w_{(C,m-1)} - w_{(C,m)}] = 0$$

Si observamos las ecuaciones de Howard, se puede ver cómo aparece un *relative state value* que no se encuentra dentro del subespacio $\hat{S}$. Se trata de $w_{(C,Q+1)}$ que aparece cuando tratamos de obtener la ecuación de Howard para $k = C$ y $m = Q$. Para obtener dicho valor será necesario recurrir a una extrapolación de algunos de los *relative state values* que forman parte del subespacio $\hat{S}$. Así el espacio de estados total a resolver es el que se muestra en la figura 7.2 y que incluye tanto el espacio de estados truncados como el estado extrapolado.

Para realizar dicha extrapolación, se ha decidido realizar un ajuste polinómico puesto que permite que las ecuaciones de Howard formen un sistema de ecuaciones lineales cerrado. Con el fin de simplificar la notación, puesto

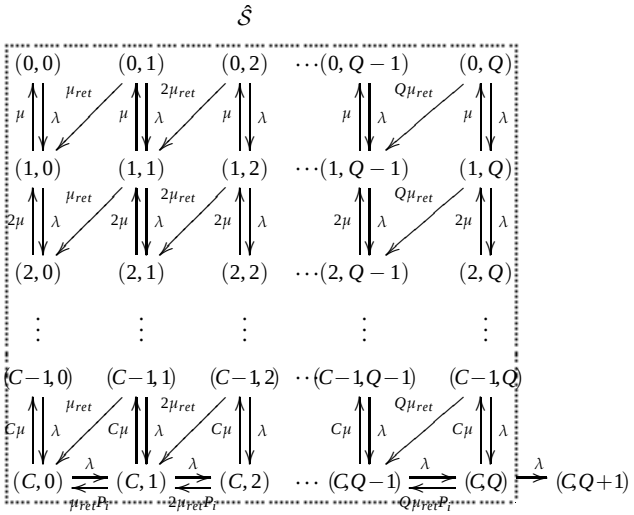


Figura 7.2: Modelo truncado para Value Extrapolation.

que todos los estados que se van a tener en cuenta para realizar la extrapolación pertenecen a la última fila del espacio de estados, es decir, $k = C$ en todos los casos, se ha decidido tomar $w_m = w_{(C,m)}$. Así, para extrapolar w_{Q+1} se ha utilizado un polinomio de grado n que interpola los $n+1$ puntos $\{(i, v_i)$ con $Q-(n+1) < i \leq Q\}$. Por ejemplo, en el caso de tomar $n = 1$, para extrapolar w_{Q+1} se hace uso de los puntos (Q, w_Q) y $(Q-1, w_{Q-1})$, obteniéndose el polinomio de interpolación $w_x = (w_Q - w_{Q-1})x + (1-Q)w_Q + Qw_{Q-1}$. Tomando $x = Q+1$ con el fin de extrapolar el punto deseado, se obtiene

$$w_{Q+1} = 2w_Q - w_{Q-1}.$$

Utilizando los polinomios de Lagrange se obtiene una expresión cerrada bastante sencilla para calcular el valor extrapolado.

$$w_{Q+1}^{(n)} = \sum_{k=0}^{n} (-1)^k \binom{n+1}{k+1} w_{Q-k} \tag{7.16}$$

En el apéndice C.3 se explica con detalle la forma de llegar a esta expresión, así como algunos ejemplos para polinomios de diferentes grados. En cuanto

Tabla 7.1: Definición de la función de coste utilizada por VE.

Probabilidad de bloqueo	P_b	$c_{(k,m)} = 1$ para $k = C, \forall m$
		$c_{(k,m)} = 0$ resto de casos
Número medio de usuarios reintentando	N_{ret}	$c_{(k,m)} = m$ $\forall k, \forall m$

al grado del polinomio a elegir, se debe considerar que, aunque en un principio aumentar el valor de n mejora la precisión del sistema, llega un momento en que no sólo no mejora, sino que puede empeorar la precisión. Asimismo, se ha de tener en cuenta que el nivel de truncación, Q, dependerá del grado del polinomio de extrapolación, de forma que $Q \geq n + 1$.

Con esto, el sistema queda totalmente definido a falta de especificar los parámetros de mérito a calcular. Para calcular estos parámetros se debe elegir la función de coste, c_s, tal que el valor medio, g, represente el parámetro de mérito que queremos calcular. En la tabla 7.1 podemos observar las diferentes funciones c_s necesarias para calcular la probabilidad de bloqueo y el número medio de usuarios reintentando.

7.3 Análisis comparativo

En este apartado se ha comparado esta aproximación con algunos de los modelos basados en el cálculo de las probabilidades de estado. En concreto, se han tomado los modelos con mejores resultados de los comparados en el apartado anterior, es decir, HM2 y AP. Adicionalmente, el escenario elegido para realizar la comparación es un escenario sin impaciencia como el que

Tabla 7.2: Orden del polinomio de extrapolación del modelo VE.

	$\mu_r = 0.001$			$\mu_r = 0.01$			$\mu_r = 0.1$			$\mu_r = 1.0$		
ρ	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
VE1	10	29	121	9	18	52	7	12	31	5	7	18
VE2	4	19	99	8	15	45	6	11	28	5	7	18
VE3	3	12	83	5	14	42	7	11	28	5	8	19
VE4	4	10	38	6	8	41	7	11	30	6	9	22
VE5	5	5	54	5	11	42	8	12	31	7	10	24
VE6	6	6	60	6	13	43	8	13	34	8	11	27
VE7	7	7	63	7	8	45	7	14	36	8	12	30
VE8	8	8	54	8	15	47	9	15	39	9	13	33

vimos en el apartado 6.2, es decir, se considera un sistema con un número de servidores de $C = 10$, y una tasa de servicio de $\mu = 1/180$. Igual que en apartados anteriores, se han considerado diferentes valores de λ, evaluándose el sistema para diferentes estados de carga. Asimismo se han considerado diferentes valores de la tasa de reintentos, concretamente se ha tomado $\mu_r = \{0.001, 0.01, 0.1, 1.0\}$. El parámetro de mérito estudiado es la probabilidad de bloqueo, calculándose el error relativo que se produce en la misma.

7.3.1 Elección de la función de ajuste

Antes poder comparar las prestaciones de VE con las de otros modelos es necesario decidir el polinomio de ajuste que se utilizará en la extrapolación. La tabla 7.2 muestra el valor de Q mínimo para obtener un error relativo en la probabilidad de bloqueo inferior a 10^{-4}. Nótese que VEx representa el uso de un polinomio de extrapolación de grado x.

Viendo la tabla no se observa una elección fácil del orden del polinomio más adecuado. Este valor varía de un escenario a otro. Sin embargo, la tendencia general es que conforme aumenta el orden del polinomio el valor de Q disminuye hasta un cierto nivel, a partir del cual el valor de Q vuelve a aumentar. En algunos casos este incremento se debe a que para usar un

Tabla 7.3: Modelo VE. Q mínima para asegurar $\epsilon \leq 10^{-4}$ en P_b.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.5	VE	4	**6**	7	6
	AP	14	10	**5**	**1**
	HM2	**3**	7	**5**	4
		$P_b = 0.0212$	$P_b = 0.0252$	$P_b = 0.0325$	$P_b = 0.0355$
0.7	VE	**10**	**8**	11	9
	AP	34	17	**7**	**1**
	HM2	17	13	9	6
		$P_b = 0.1252$	$P_b = 0.1541$	$P_b = 0.2002$	$P_b = 0.2186$
0.9	VE	**38**	41	30	22
	AP	112	**32**	**10**	**1**
	HM2	88	38	22	12
		$P_b = 0.4692$	$P_b = 0.5355$	$P_b = 0.6299$	$P_b = 0.6633$

polinomio de grado x es necesario utilizar un modelo con $Q \geq x$; como se puede ver, por ejemplo, en el escenario $\mu_r = 0.001$ y $\rho = 0.5$ a partir de VE4. Aunque el orden del polinomio no va a afectar al coste computacional de resolución se ha decidido utilizar polinomios de extrapolación de orden 4 para todos los escenarios. De este modo, de ahora en adelante se empleará VE4, denotándolo simplemente como VE.

7.3.2 Comparación con otros métodos

Elegido el polinomio de ajuste del modelo VE, vamos a comparar las prestaciones de este modelo con las de los modelos AP y HM2, tanto en términos de complejidad —es decir, valor de Q mínimo para garantizar una precisión— como de coste computacional. En la Tabla 7.3 se muestran los valores de Q mínimos necesarios para obtener un error relativo inferior a 10^{-4} en la probabilidad de bloqueo.

Los resultados muestran que VE obtiene los mejores resultados cuando la tasa de reintentos es muy baja, mientras que en el resto de casos se ve

Tabla 7.4: Modelo VE. Tiempo de resolución para la Q mínima.

ρ		$\mu_r = 0.001$	$\mu_r = 0.01$	$\mu_r = 0.1$	$\mu_r = 1.0$
0.5	VE	**0.0018**	**0.0018**	**0.0020**	**0.0018**
	AP	0.491	0.3501	0.1782	0.0472
	HM2	0.0317	0.0096	0.0080	0.0070
0.7	VE	**0.0018**	**0.0040**	**0.0036**	**0.0025**
	AP	1.1855	0.6001	0.2481	0.0418
	HM2	0.0309	0.0201	0.0133	0.0104
0.9	VE	**0.0507**	**0.0394**	**0.0252**	0.0195
	AP	3.8879	1.1180	0.3550	0.0469
	HM2	0.1316	0.0483	0.0365	**0.0194**

superado por AP y/o HM2. Para el resto de casos, el modelo que obtiene mejores resultados es AP. No obstante, tal y como hemos comentado con anterioridad, este modelo aunque presenta muy buenos resultados en cuanto a precisión resulta lento a la hora de resolver. Es por ello que comparamos también los resultados de VE con los modelos AP y HM2 en términos de coste computacional. En la tabla 7.4 se muestra el tiempo empleado para resolver los diferentes modelos con la Q necesaria para garantizar un error de 10^{-4}.

Los mejores resultados en cuanto a tiempo de cómputo los obtiene VE, que resulta siempre el algoritmo más rápido tal y como se observa en la tabla 7.4. Tanto VE como HM2 consiguen tiempos de resolución por debajo del segundo en todos los casos estudiados, de forma que la diferencia entre ellos es despreciable desde el punto de vista humano. Esto no ocurre con AP, que requiere tiempos de computo elevados, en especial cuando la carga del sistema es elevada y la tasa de reintentos baja.

Por último en la figura 7.3 se relaciona la precisión obtenida con el tiempo de cómputo de los diferentes modelos conforme varía Q para un escenario con $\mu_r = 0.1$ y $\rho = 0.7$. Como se puede observar el modelo VE ofrece la mejor compromiso entre precisión y tiempo de todos los modelos comparados.

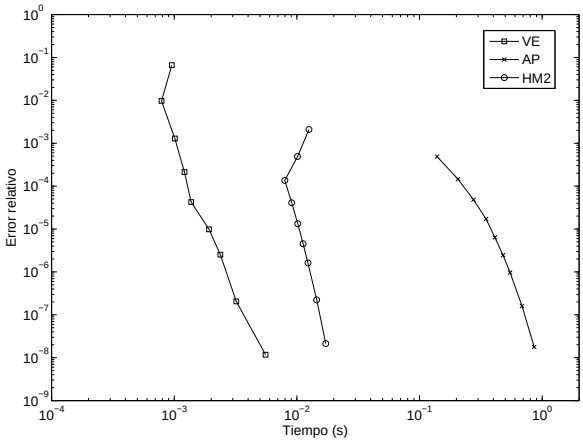


Figura 7.3: Error relativo para P_b en función del tiempo de cálculo.

7.4 Conclusiones

En esta sección se ha desarrollado un nuevo paradigma para la resolución de sistemas de reintentos como es *Value Extrapolation*. La principal característica de este método es que no se basa en el cálculo de las probabilidades de estado en régimen permanente, sino en una nueva métrica denominada *relative state values*, que aparece cuando se considera el sistema como un MDP. Los resultados obtenidos nos permiten concluir que el modelo VE aunque no consigue mejorar en términos de precisión para todos los escenarios, sí que asegura una resolución muy rápida de cualquier escenario que se pretenda estudiar. Pero además, el hecho de no basarse en la obtención de las probabilidades de estado le confiere una versatilidad que puede resultar muy útil en la resolución de sistemas más complejos.

Capítulo 8

Sistemas de reintentos en redes celulares

Los reintentos pueden, en condiciones de sobrecarga generar un efecto de bola de nieve en los procesos de llegada que degrade fuertemente la calidad de servicio percibida por los usuarios. Tran-Gia y Mandjes demuestran, en [TGM97], que el efecto de los reintentos en redes móviles celulares no es despreciable. Esto hace necesario realizar un modelado preciso del comportamiento de los usuarios, con el fin diseñar correctamente el sistema. Dicho modelado deberá incorporar la existencia de reintentos.

Aunque en redes fijas el estudio de los reintentos ha sido profusamente estudiado y podemos encontrar las primeras referencias ya en los años 70 para modelar el comportamiento de los usuarios en las redes de telefonía básica [JS70], el problema de los reintentos en redes móviles no ha sido tan ampliamente estudiado y tendremos que esperar hasta 1997 con [TGM97], en que se demuestra la necesidad de considerar la existencia de reintentos en redes de celulares. A partir de este trabajo aparecen diversas referencias que tratan de modelar el efecto de reintentos en este tipo de redes, entre los más destacados podemos encontrar los modelos propuestos en el propio [TGM97], pero también [MCL+01, AL02]. En [TGM97] se presentan dos modelos, el primero con población finita y sin control de admisión y el segundo con población infinita y una política de control de admisión basada

en la reserva de recursos. Otro de los trabajos más conocidos es el propuesto por Marsan et al en [MCL+01] donde se presenta un modelo aproximado para la resolución de sistemas de reintentos con un único tipo de servicio (que incluirá peticiones nuevas y de *handover*), basado en convertir el espacio de estados del modelo original en otro simplificado mediante la agrupación de estados. Destaca en este modelo la posibilidad de diferenciar los reintentos de las sesiones nuevas de aquellos realizados por los traspasos mediante la definición de dos órbitas de reintento diferentes. Por último mencionar que todo el desarrollo se ha realizado tanto para sistemas de población finita como infinita. Por otro lado, Alfa y Li en [AL02] consideran un modelo en que el proceso de llegada no es de Poisson, sino que se modela como un *Markovian Arrival Process*, MAP [Neu81]. Asimismo, para el tiempo medio entre reintentos hacen uso de una distribución *phase-type* [LR99]. Más recientemente, aparecen los trabajos de Roszik [RSK05] y Chakravarthy [CKJ06]. En [RSK05] se desarrolla un modelo para fuentes infinitas basado en los modelos [TGM97] y [MCL+01], que permite analizar modelos más complejos y que consigue reducir el espacio a un conjunto finito limitando el número máximo de usuarios reintentado. El modelo propuesto en [CKJ06] introduce una característica novedosa como es el hecho de que, tras terminar un servicio, el servidor tomará, con una determinada probabilidad, uno de los usuarios esperando en la órbita de reintentos. Otra de las características de este modelo es que considera que las llegadas de usuarios tienen una distribución MAP en lugar de poissoniana como es habitual.

Todos los modelos anteriores consideran que el reintento se debe al comportamiento humano. Pero adicionalmente, en las redes celulares podemos encontrar reintentos debido al propio funcionamiento de la red y su estructura celular. Destacar el estudio realizado por Eklundh [Ekl86], en el que se introduce el concepto de *directed retrial*. En este trabajo se plantea la posibilidad de que, para evitar la congestión en caso de que no existan recursos suficientes en la célula actual, el sistema pueda buscar recursos en las células vecinas. Tomando este trabajo como punto de partida, aparece la propuesta de Onur et al en [ODEa02], que plantea un modelo en el que no sólo se con-

sideran reintentos debido al comportamiento humano (a los que denomina remarcados), sino también reintentos automáticos que realiza la propia red cuando una petición nueva o un remarcado quedan bloqueados, sin que el usuario se percate de lo que está ocurriendo.

En este capítulo se introducen las características de los reintentos en el caso particular de las redes móviles celulares, haciendo hincapié en las diferencias que podemos encontrar entre los reintentos producidos por el comportamiento humano y aquellos controlados por la red. Las diferencias entre estos dos tipos de reintentos darán lugar a la necesidad de modelos del sistema mucho más complejos, tal y como observaremos en este capítulo. Por último aprovecharemos este capítulo para observar las consecuencias de ignorar la existencia de reintentos en la red o de no considerarlos como tal.

8.1 Reintentos en redes móviles celulares

En una red celular podemos encontrar reintentos debido al bloqueo de una petición de recursos por parte de un usuario. Es decir, cuando un usuario llega al sistema y encuentra todos los servidores ocupados, abandonará el área de servicio para volver a intentar acceder al sistema tras un cierto tiempo de espera. Puesto que estos reintentos dependen exclusivamente del comportamiento humano, se considera que el tiempo entre reintentos es aleatorio y que la persistencia en el remarcado dependerá de la paciencia de los usuarios. A este tipo de reintento le denominaremos *remarcado* —del inglés *redial*— a partir de ahora.

Sin embargo, debido a la naturaleza celular de la arquitectura, así como a la movilidad de los usuarios, en este tipo de redes existen otro tipo de reintentos [Ekl86]. Las redes celulares están divididas en diferentes áreas de servicio, denominadas células, estando cada una de ellas servida por una única estación base y con una cantidad de recursos (servidores) que se emplearán para satisfacer las demandas de los usuarios que se encuentren en su área geográfica. Por otra parte los usuarios de estas redes, incluso aquellos que tienen

una comunicación activa, se mueven entre las diferentes células produciendo *handovers*. Cuando un usuario activo pasa de ser servido por una célula a otra diferente, se ejecuta un proceso de *handover* que se encarga de atender a dicho usuario y asignarle los recursos necesarios en la célula destino del *handover* y liberar los recursos que venía utilizando en la célula origen. Así, los usuarios —terminales móviles— que traten de realizar un *handover* entre células adyacentes y queden bloqueadas, pueden reintentar el acceso mientras el usuario se encuentre en la zona de solape entre las dos células implicadas en el *handover*. Este tipo de reintento, al que denominaremos *reintento automático* —en inglés *retrial*, en contraposición con los *redials*—, se realiza de forma automática por parte de la red y el terminal móvil, sin el conocimiento del usuario. En este caso el sistema reintentará, bien hasta que encuentre suficientes recursos libres en la célula destino para seguir cursando la sesión, o bien hasta que el usuario —terminal móvil— abandone el área de solape. En este último caso la sesión se cortará mientras estaba en curso. De lo contrario, la llamada continuará su curso sin que el usuario perciba ninguna interrupción. Este tipo de reintentos está incluido en el estándar de GSM, permitiendo realizar un número máximo de reintentos consecutivos [ODEa02].

Se observa claramente que las características de remarcados y reintentos automáticos van a ser diferentes, con lo que resulta necesario tratarlos de forma diferenciada. Como consecuencia, el modelado de sistemas celulares con reintentos debe considerar dos órbitas de reintentos diferentes, una para cada uno de los tipos de reintentos existentes en este tipo de red.

Nótese que en algunos modelos, puesto que el tiempo entre reintentos consecutivos debidos a un *handover* es mucho menor que el tiempo de servicio y que el tiempo entre la llegada de peticiones consecutivas de servicio, estos reintentos se han modelado mediante una cola con impaciencia [Bar04]. En estos casos, el tiempo de permanencia en dicha cola representa el periodo de permanencia en el área de solape.

8.2 Modelo del sistema

Se considera una red móvil celular homogénea donde todas las células son estadísticamente idénticas e independientes, con lo que para analizar la red es suficiente con analizar una única célula [Gué87]. Se considera, asimismo, la utilización de una asignación fija de recursos de forma que cada célula disponga de un número C fijo de recursos. El significado físico de una unidad de recursos depende de la tecnología que se utilice para implementar la interfaz radio. Por otro lado, a la hora de modelar la red de comunicaciones, el número de recursos de los que dispone la célula se traduce por el número de servidores del que dispondrá el sistema. Además, se considera un sistema monoservicio en el que se distingue entre usuarios nuevos y *handovers*. El hecho de tratarse de un sistema monoservicio nos permite considerar, sin pérdida de generalidad, que cada usuario que accede al sistema ocupa un único recurso.

Con estas características, a la célula llegan dos flujos de peticiones: peticiones de sesiones nuevas y de *handovers* desde células adyacentes. Se considera que ambos flujos son de Poisson con tasas λ_n y λ_h, respectivamente, con $\lambda = \lambda_n + \lambda_h$. Para determinar el valor de λ_h la solución habitual supone que el flujo de *handovers* entrantes en la célula es igual al flujo de salida de *handovers* [MCL+99].

Cuando llega una petición al sistema, si existen recursos suficientes para atenderla, esta se acepta y se le asignan los recursos necesarios. Si no es así — si no hay recursos suficientes disponibles — la sesión se bloqueará. El tiempo de servicio se asume regido por una distribución exponencial de tasa μ_s. Asimismo se considera que el tiempo de residencia en la célula también estará distribuido exponencialmente con tasa μ_{re}. Por lo que, debido a la propiedad de memoria nula de la distribución exponencial, el tiempo de ocupación de los recursos también está distribuido exponencialmente con tasa $\mu = \mu_s + \mu_{re}$.

Cuando la petición es bloqueada por el sistema, esta reintentará con cierta probabilidad o abandonará el sistema con la probabilidad complementaria.

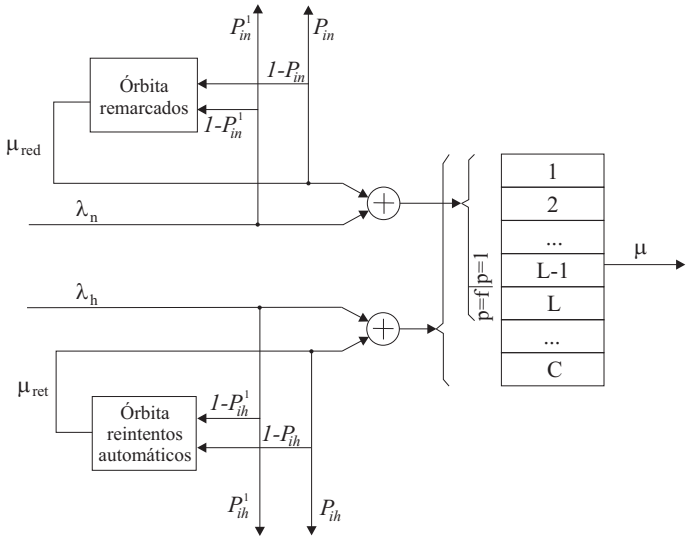


Figura 8.1: Modelo sistema de reintentos con dos órbitas.

Si se trata de una petición de servicio nueva, ésta se unirá a la órbita de remarcados con probabilidad $(1 - P_{in}^1)$ o abandonará el sistema con probabilidad P_{in}^1, según se observa en la figura 8.1. Los usuarios en la órbita de remarcados reintentan el acceso a los servidores tras un tiempo exponencialmente distribuido de tasa μ_{red}. En caso de que el remarcado sea bloqueado, volverá a la órbita de reintentos con probabilidad $(1 - P_{in})$ o abandonará el sistema con probabilidad P_{in}. De este modo se distingue el primer reintento de los siguientes. Para el caso de los reintentos automáticos tenemos un comportamiento paralelo para el cual se definen los parámetros P_{ih}^1, P_{ih} y μ_{ret}. Nótese que si se toma $P_{ih}^1 = 0$, en caso de bloqueo, como mínimo se efectuará un reintento. Con dicha configuración, si el sistema está lo suficientemente cargado para poder considerar que la probabilidad de que un reintento consiga acceso a los servidores es cero, el tiempo transcurrido desde el primer reintento hasta que el usuario abandona el sistema, μ_r', se puede calcular,

tomando transformadas de Laplace, como:

$$\frac{\mu_{ret}}{s+\mu_{ret}}P_{ih} + \left(\frac{\mu_{ret}}{s+\mu_{ret}}\right)^2(1-P_{ih})P_{ih} + \left(\frac{\mu_{ret}}{s+\mu_{ret}}\right)^3(1-P_{ih})^2P_{ih} + \ldots =$$
$$\frac{\mu_{ret}}{s+\mu_{ret}}P_{ih}\frac{1}{1-\frac{\mu_{ret}}{s+\mu_{ret}}(1-P_{ih})} = \frac{\mu_{ret}P_{ih}}{s+\mu_{ret}P_{ih}} = \frac{\mu'_r}{s+\mu'_r},$$

es decir $\mu'_r = P_{ih}\mu_{ret}$. O dicho de otro modo, μ'_r será la suma de X variables aleatorias exponenciales independientes e idénticamente distribuidas de media μ_{ret}^{-1}. Puesto que la variable aleatoria X sigue una distribución geométrica de media $1/P_{ih}$, el tiempo entre que se produce el primer intento hasta que el usuario abandona presentará una distribución exponencial de tasa $\mu'_r = P_{ih}\mu_{ret}$. Conforme a estas características, el modelo considera el tiempo de residencia en el área de solape como una variable aleatoria exponencial de tasa μ'_r. Nótese que, aunque en [RGS98] y [PCG02a] se concluye que el tiempo de residencia en el área de solape no es exponencial, esta suposición tiene un bajo impacto en los parámetros de mérito, tal y como se demuestra en [PCG02b].

Se observa también en la figura 8.1 que el acceso a recursos por parte de peticiones nuevas y *handovers* no es el mismo. En general, el bloqueo de una petición de servicio, tal y como ocurre en los remarcados, degrada menos la calidad de servicio experimentada por los usuarios que el bloqueo de un *handover*. En el caso de que se trate una sesión de tiempo real (*streaming*), el bloqueo del *handover* produce la terminación abrupta de una sesión en curso, lo que resulta más molesto a un usuario que el hecho de retrasar el inicio de una sesión nueva. En el caso de que se tratase de tráfico elástico [BR03] el efecto aún sería peor, puesto que la información transmitida hasta ese momento sería inútil para el extremo receptor. Puesto que la pérdida de un *handover* es más problemática que el bloqueo de una petición nueva, es muy habitual que los sistemas de comunicaciones presenten algún mecanismo de control de admisión que asegure que la pérdida de *handovers* ocurra con menor frecuencia que la de peticiones nuevas. Así se da cierta prioridad a los *handovers*. En este caso se ha optado por un mecanismo denominado *Fractional Guard Channel* (FGC) [RTN97]. Este mecanismo queda definido mediante

un único parámetro t ($0 \leq t \leq C$), de forma que las peticiones nuevas se aceptan, con probabilidad 1, si el número de servidores ocupados es menor que $L = \lfloor t \rfloor$ y con probabilidad $f = t - L$ si el número de servidores ocupados es, exactamente, L. Si hay más de L recursos ocupados no se aceptarán más peticiones nuevas. Mientras que los *handovers* serán aceptados siempre que queden recursos libres. Como se muestra en la figura 8.1 los remarcados obtendrán el mismo tratamiento que las peticiones nuevas, mientras que los reintentos automáticos obtendrán el tratamiento de los *handovers*.

8.2.1 Naturaleza determinista de los reintentos automáticos

En sistemas reales, el tiempo entre reintentos automáticos así como el máximo número de reintentos por petición toman un valor determinista [ODEa02]. A la hora de modelar, sin embargo, por motivos de simplicidad, se ha utilizado un tiempo entre reintentos exponencial y una distribución geométrica para el número máximo de reintentos.

La suposición de exponencialidad en el tiempo entre reintentos consecutivos no tiene consecuencias apreciables en el estudio de los parámetros de mérito de la red. Esto no ocurre con el número máximo de reintentos, donde los resultados varían sustancialmente según el tipo de distribución considerada. En este sentido se ha realizado una aproximación que permite comparar el comportamiento de una distribución determinista con el de una geométrica.

Para ello vamos a igualar el número medio de reintentos que se obtienen con ambas distribuciones, la geométrica y la determinista. Esto difiere del hecho de que ambas distribuciones posean la misma media, ya que las distribuciones se refieren al número máximo de reintentos, no a su número medio. Se parte de un parámetro d que indique el número máximo de intentos —considerando el intento inicial— que se permiten cuando se considera una distribución determinista. Definido este valor, y definiendo como q la probabilidad de bloqueo de un reintento —nótese que esta q no tiene porqué ser

igual a la probabilidad de bloqueo de los *handovers*, P_b^h—, se define el número medio de reintentos por usuario para el caso de utilizar una distribución geométrica como:

$$u_h^{Geo} = \sum_{n\geq 1} nP_b^h(1-P_{ih}^1)((1-P_{ih})q)^{n-1}(qP_{ih}+(1-q)) = \\ = \frac{P_b^h(1-P_{ih}^1)qP_{ih}}{(1-(1-P_{ih}q)^2)} + \frac{P_b^h(1-P_{ih}^1)(1-q)}{(1-(1-P_{ih})q)^2} = \frac{(1-P_{ih}^1).P_b^h}{1-(1-P_{ih})q}, \quad (8.1)$$

donde el primer término tras el segundo signo de igualdad representa el número medio de reintentos que acaba en abandono y el segundo término el número medio de reintentos que acaba en aceptación.

Mientras que para el caso determinista tendremos:

$$u_h^D = \quad (1-q)P_b^h[1+2q+3q^2+\ldots+(d-1)q^{d-2}] + dP_b^h q^{d-1} = P_b^h\frac{1-q^d}{1-q}, \quad (8.2)$$

donde el primer término tras el primer signo de igualdad indica el número medio de reintentos que acaba en una aceptación y el segundo término indica el número medio de reintentos que acaba en un abandono.

Si suponemos que q y P_b^h toman aproximadamente el mismo valor en ambos casos, se iguala u_h^D con u_h^{Geo}, de forma que despejando se obtiene:

$$P_{ih} = \frac{1-q}{q(1-q^d)}(q^d - P_{ih}^1). \quad (8.3)$$

Para un valor dado de d, usando (8.3) junto con las expresiones necesarias para calcular P_b^h así como $u_h = u_h^D = u_h^{Geo}$, se puede calcular el valor de P_{ih} que asegura que los valores medios de reintentos por usuario son los mismos que los se obtendrían con una distribución determinista con un número máximo de reintentos igual a d. Dado un valor de d determinado, para obtener el valor de P_{ih} es necesario un proceso iterativo del estilo:

1. damos valor a $q = P_b^h \approx 1$

2. calculamos la P_{ih} haciendo uso de (8.3)

3. con el valor calculado de P_{ih} se resuelve el sistema obteniendo los nuevos valores de P_b^h y u_h

4. si la diferencia relativa entre el valor de P_b^h calculado en esta iteración y el anterior es mayor que una determinada tolerancia volver al punto (2), parar en caso contrario.

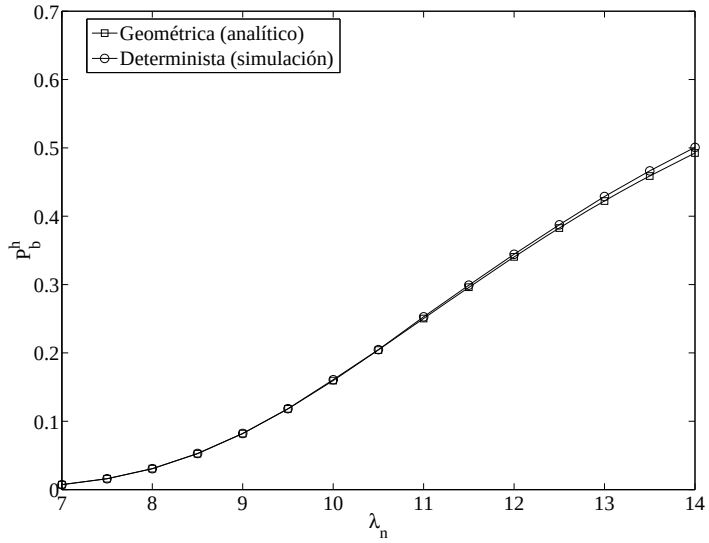
Esta aproximación, tal y como se muestra en la figura 8.2, consigue unos excelentes resultados en el análisis de prestaciones.

8.3 Resolución sistemas de dos órbitas

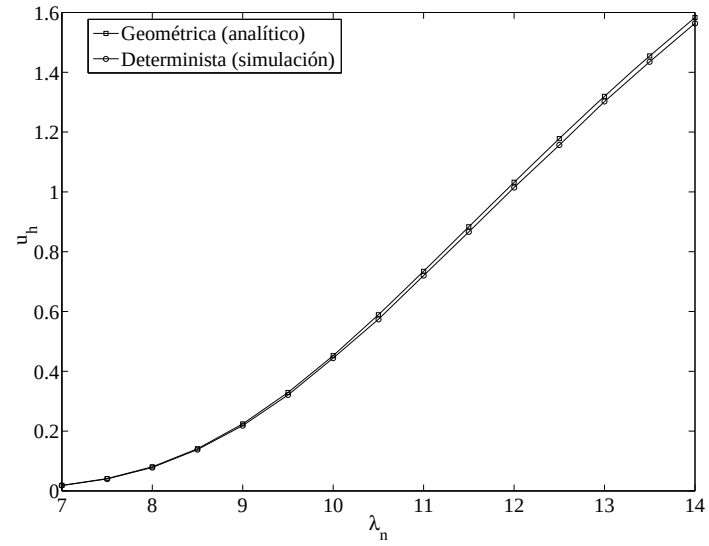
El modelo de dos órbitas se representa como una CTMC tridimensional (k, m, s), donde k es el número de servidores ocupados, m el número de usuarios en la órbita de remarcados y s el de usuarios en la órbita de reintentos automáticos. En este caso nos encontramos ante una representación de la cadena de Markov con dos dimensiones infinitas, $\{0, \ldots, C\} \times \mathbb{Z}_+ \times \mathbb{Z}_+$, y la no-homogeneidad de las mismas.

Al igual que en el caso del sistema de reintentos de una órbita, es necesario recurrir a modelos aproximados para la resolución de este tipo de modelos. Sin embargo, el hecho de presentar dos órbitas infinitas va a limitar considerablemente el número de soluciones posibles. En este caso no es posible utilizar modelos truncados generalizados en ambas dimensiones. Así, será necesario recurrir a un modelo truncado en al menos una de las dos dimensiones infinitas. Mientras que en la segunda se puede optar por una un modelo truncado o truncado generalizado. En este caso se ha optado por utilizar el modelo FM en ambas órbitas por ser el mejor de los modelos truncados estudiados.

Este modelo estaba basado en la agregación de estados, de forma que se reduce el espacio de estados a un conjunto finito de estados mediante la



(a) Probabilidad de bloqueo de *handovers*



(b) Número medio de usuarios reintentando

Figura 8.2: Comparación distribución geométrica con determinista.

Tabla 8.1: Tasas de transición del modelo FM en el sistema de 2 órbitas.

Transición	Condición		Tasa
$(k,m,s) \to (k+1,m,s)$	$0 \le k \le L-1$	$m < Q_n$ & $s < Q_h$	λ
		$m < Q_n$ & $s = Q_h$	$\lambda + \beta_h$
		$m = Q_n$ & $s < Q_h$	$\lambda + \beta_n$
		$m = Q_n$ & $s = Q_h$	$\lambda + \beta_n + \beta_h$
	$k = L$	$m < Q_n$ & $s < Q_h$	$\lambda_h + f\lambda_n$
		$m < Q_n$ & $s = Q_h$	$\lambda_h + \beta_h + f\lambda_n$
		$m = Q_n$ & $s < Q_h$	$\lambda_h + f(\beta_n + \lambda_n)$
		$m = Q_n$ & $s = Q_h$	$\lambda_h + \beta_h + f(\beta_n + \lambda_n)$
	$L < k \le C$	$m < Q_n$ & $s < Q_h$	λ_h
		$m < Q_n$ & $s = Q_h$	$\lambda_h + \beta_h$
		$m = Q_n$ & $s < Q_h$	λ_h
		$m = Q_n$ & $s = Q_h$	$\lambda_h + \beta_h$
$(k,m,s) \to (k+1,m,s-1)$	$0 \le k \le C-1$	$1 \le s \le Q_h - 1$	$s\mu_{ret}$
		$s = Q_h$	α_h
$(k,m,s) \to (k,m,s-1)$	$k = C$	$1 \le s \le Q_h - 1$	$s\mu_{ret}P_{ih}$
		$s = Q_h$	$\alpha_h P_{ih}$
$(k,m,s) \to (k+1,m-1,s)$	$0 \le k \le L-1$	$1 \le m \le Q_n - 1$	$m\mu_{red}$
		$m = Q_n$	α_n
	$k = L$	$1 \le m \le Q_n - 1$	$m\mu_{red}f$
		$m = Q_n$	$\alpha_n f$
$(k,m,s) \to (k,m-1,s)$	$k = L$	$1 \le m \le Q_n - 1$	$m\mu_{red}(1-f)P_{in}$
		$m = Q_n$	$\alpha_n(1-f)P_{in}$
	$L < k \le C$	$1 \le m \le Q_n - 1$	$m\mu_{red}P_{in}$
		$m = Q_n$	$\alpha_n P_{in}$
$(k,m,s) \to (k-1,m,s)$	$1 \le k \le C$		$k\mu$
$(k,m,s) \to (k,m,s+1)$	$k = C$		$\lambda_h(1-P_{ih}^1)$
$(k,m,s) \to (k,m+1,s)$	$k = L$		$\lambda_n(1-P_{in}^1)(1-f)$
	$L < k \le C$		$\lambda_n(1-P_{in}^1)$
Nota:	$\alpha_n = M_n\mu_{red}(1-p_n)$, $\beta_n = M_n\mu_{red}p_n$		
	$\alpha_h = M_h\mu_{ret}(1-p_h)$, $\beta_h = M_h\mu_{ret}p_h$.		

agregación de todos los estados en los que la ocupación de las órbitas supere un determinado valor. En concreto, Q_n (Q_h) define la ocupación, número de usuarios en la órbita de remarcados (reintentos automáticos) a partir de la cual se agregan los estados. A parte de estos dos parámetros, y debido a la agregación, aparecerán dos parámetros más para cada órbita. El parámetro M_n denota el número medio de usuarios en la órbita de remarcados cuando hay, por lo menos, Q_n usuarios en la misma, es decir, $M_n = E[m|m \geq Q_n]$. Asimismo denotamos como p_n a la probabilidad de que tras un remarcado exitoso —es decir, se consigue acceder a los servidores— el número de usuarios en la órbita de remarcados no tome valores por debajo de Q_n. Para la órbita de reintentos automáticos los parámetros M_h y p_h se definen de forma análoga. Como resultado de la agregación el espacio de estados a resolver tiene la forma:

$$S = \{(k, m, s) : 0 \leq k \leq C; 0 \leq m \leq Q_n; 0 \leq s \leq Q_h\},$$

donde los estados de la forma $(\cdot, Q_n, \cdot)$ representan las situaciones en que por lo menos existen Q_n usuarios en la órbita de remarcados. De forma similar, los estados de la forma $(\cdot, \cdot, Q_h)$ representan los estados con Q_h o más usuarios en la órbita de reintentos automáticos. Podemos observar las tasas de transición de este modelo en la tabla 8.1.

Para obtener las probabilidades de estado en régimen permanente de este sistema es necesario conocer los valores de los parámetros de la aproximación, M_n, p_n, M_h y p_h. Para obtener estos parámetros se hace uso de los flujos de balance de probabilidad junto con el hecho de que la tasa de bloqueo de peticiones nuevas que reintentan es igual a la suma de las tasas de reintentos exitosos y abandonos. Las expresiones de los parámetros de la aproximación, expresadas en función de las probabilidades de estado, vendrán dadas por las siguientes expresiones:

$$p_h = \frac{\sum_{m=0}^{Q_n} \pi(C, m, Q_h)}{\sum_{m=0}^{Q_n} \left[\pi(C, m, Q_h) + \pi(C, m, Q_h - 1)\right]}$$

$$M_h = \frac{\lambda_h(1 - P_{ih}^1)\left(\sum_{m=0}^{Q_n} \left[(\pi(C, m, Q_h) + \pi(C, m, Q_h - 1)\right]\right)}{\mu_{ret}\left(\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n} \pi(k, m, Q_h) + P_{ih}\sum_{m=0}^{Q_n} \pi(C, m, Q_h)\right)}$$

$$p_n = \frac{\zeta_1}{\zeta_2} \quad ; \qquad M_n = \frac{\lambda_n(1 - P_{in}^1)\zeta_2}{\mu_{red}\zeta_3}$$

donde

$$\zeta_1 = \sum_{k=L+1}^{C}\sum_{s=0}^{Q_h} \pi(k, Q_n, s) + (1 - f)\sum_{s=0}^{Q_h} \pi(L, Q_n, s)$$

$$\zeta_2 = \sum_{k=L+1}^{C}\sum_{s=0}^{Q_h} \left[\pi(k, Q_n - 1, s) + \pi(k, Q_n, s)\right] + (1 - f)\sum_{s=0}^{Q_h} \left[\pi(L, Q_n - 1, s) + \pi(L, Q_n, s)\right]$$

$$\zeta_3 = \sum_{k=0}^{L-1}\sum_{s=0}^{Q_h} \pi(k, Q_n, s) + f\sum_{s=0}^{Q_h} \pi(L, Q_n, s) + (1 - f)P_{in}\sum_{s=0}^{Q_h} \pi(L, Q_n, s) + P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h} \pi(k, Q_n, s)$$

En el Apéndice C.4 se observa el procedimiento seguido para llegar a estas expresiones.

Las ecuaciones de balance, junto con la ecuación de normalización y las ecuaciones de los parámetros de la aproximación forman un sistema de ecuaciones no lineales que se resolverá utilizando un proceso iterativo. Este proceso parte de unos valores iniciales para los parámetros de mérito, en concreto $p_n = p_h = 0$, $M_n = Q_n$ y $M_h = Q_h$. A partir de estos valores se calculan las probabilidades de estado en régimen permanente del sistema que, a su vez,

darán lugar a los nuevos valores de los parámetros p_n, p_h, M_n y M_h . Este proceso se repetirá hasta que el error entre dos iteraciones consecutivas sea menor que un determinado $\epsilon = 10^{-4}$ para todos los parámetros.

Los parámetros de mérito más comúnmente empleados en las redes celulares son las probabilidades de bloqueo tanto de sesiones nuevas, P_b^n, como de *handovers*, P_b^h. De manera adicional, también se ha hecho uso de la probabilidad de que una sesión resulte interrumpida debido a la imposibilidad de realizar alguno de los *handovers*, probabilidad a la que se le denomina de terminación forzosa, P_{ft}. En un sistema con reintentos, dicha probabilidad se calcula a partir de la probabilidad de no servicio de *handovers*, P_{ns}^h), es decir, la probabilidad de que un *handover* y todos sus reintentos automáticos asociados resulten bloqueados. Las expresiones de dichos parámetros son:

$$P_b^n = \sum_{k=L+1}^{C} \sum_{m=0}^{Q_n} \sum_{s=0}^{Q_h} \pi(k,m,s) + (1-f) \sum_{m=0}^{Q_n} \sum_{s=0}^{Q_h} \pi(L,m,s)$$

$$P_b^h = \sum_{m=0}^{Q_n} \sum_{s=0}^{Q_h} \pi(C,m,s)$$

$$P_{ns}^h = \frac{\mu_{ret}}{\lambda_h} P_{ih} \Big[\sum_{m=0}^{Q_n} \sum_{s=0}^{Q_h-1} s\pi(C,m,s) + M_h \sum_{m=0}^{Q_n} \pi(C,m,Q_h) \Big] + P_{ih}^1 \sum_{m=0}^{Q_n} \sum_{s=0}^{Q_h} \pi(C,m,s)$$

$$P_{ft} = \frac{N_H P_{ns}^h}{1 + N_H P_{ns}^h} \qquad \text{con} \quad N_H = \frac{\mu_{re}}{\mu_s}$$

8.3.1 Precisión del modelo desarrollado

Como primer paso se ha evaluado la precisión del modelo propuesto para la resolución de un sistema de dos órbitas como el que podemos encontrar en una red celular. Dicha precisión se ha expresado como función de los parámetros Q_n y Q_h. Así, se ha calculado el error relativo para una función indicadora, I, y dados dos valores para Q_n y Q_h, como:

$$\epsilon_I(Q_n, Q_h) = \Big| \frac{I(Q_n+1, Q_h+1)}{I(Q_n, Q_h)} - 1 \Big| < 10^{-4}$$

Para todos los experimentos llevados a cabo, tanto en este apartado como en apartados posteriores, y a menos que se indique lo contrario, se han empleado la siguiente configuración: $C = 32$, $N_H = \mu_{re}/\mu_s = 2$, $\mu = \mu_{re} + \mu_s = 1$, $t = 31$, $\mu_{red} = 20$, $P_{in}^1 = P_{ih}^1 = 0$, $P_{ih} = 0.2$ $\mu_r' = 10\mu_r$ y $\mu_{ret} = 100/3$. Con el fin de simplificar los estudios realizados en este capítulo y asegurar que los diferentes modelos estudiados sean comparables se ha tomado $\lambda_h = 2\lambda_n$, en lugar de utilizar el balance de tasas entrantes y salientes de la célula.

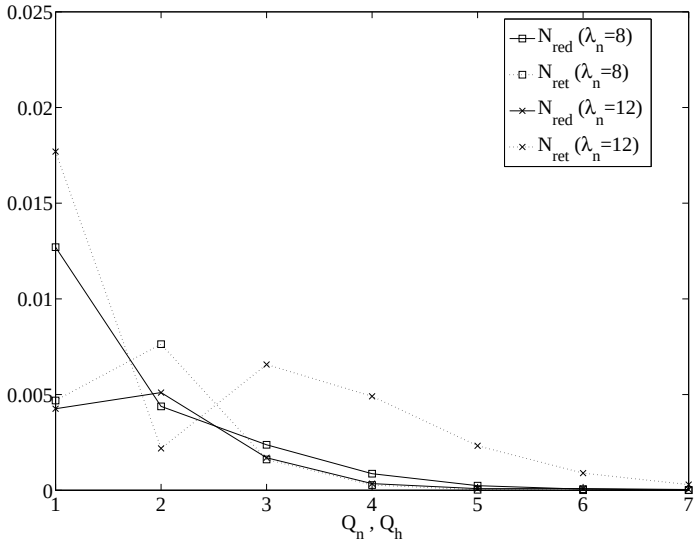


Figura 8.3: Precisión del modelo de dos órbitas.

En la figura 8.3 se muestra el error relativo estimado como función de $Q = Q_n = Q_h$ empleando como indicadores N_{red} y N_{ret}. Se ha evaluado el comportamiento del modelo para dos tasas de llegada diferentes, $\lambda_n = 8$ y $\lambda_n = 12$, con el objetivo de estudiar el sistema con baja carga y altamente cargado. Como cabría esperar, excepto para una pequeña fase transitoria, se observa que el valor de $\epsilon_I(Q_n, Q_h)$ disminuye conforme aumentan Q. También se observa como, para garantizar una determinada precisión, una carga

alta requiere un valor de Q mayor que una carga baja. Sin embargo, las curvas muestran que se pueden conseguir buenas prestaciones con valores relativamente bajos de Q.

8.4 Impacto de los reintentos en una red de comunicaciones

Debido a la escasez de recursos, así como a la movilidad de los dispositivos en las redes móviles celulares, las peticiones de recursos pueden quedar bloqueadas durante la petición de un nuevo servicio e incluso se pueden abortar sesiones en curso debido a la falta de recursos durante un proceso de *handover*. Cuando surge alguna de estas situaciones, es habitual que los usuarios, o incluso la propia red, reintente el acceso a los recursos. No obstante, a la hora de diseñar la red, ha sido común que el operador no considere la existencia de reintentos puesto que la red no es capaz de distinguir entre un primer intento y un reintento. Con la imposibilidad de distinguir reintentos de primeros accesos, en el diseño de red se han considerado dos opciones, bien ignorar la existencia de los reintentos —modelo al que denominaremos Sin Remarcados—, o bien introducir una carga adicional a la ya esperada que introduzca el efecto de los reintentos —modelo que se referenciará como Modelo Simplificado—. En esta sección se evalúan estas dos soluciones.

Nos hemos centrado en el caso de los remarcados puesto que consideramos que tendrán un impacto mayor en las prestaciones de la red que los reintentos automáticos. Nótese que la red se diseña para asegurar una probabilidad de bloqueo de *handovers* muy baja, con lo que habrá muy pocos reintentos automáticos. Es por ello que nos hemos centrado en el estudio del error que se produce en el diseño cuando no se consideran los remarcados. De este modo, se evalúa el efecto de hacer uso de estas dos soluciones únicamente para el caso de los remarcados. Para ello, se modela el sistema tal y como se observa en la figura 8.4. En dicho modelo desaparece la órbita de

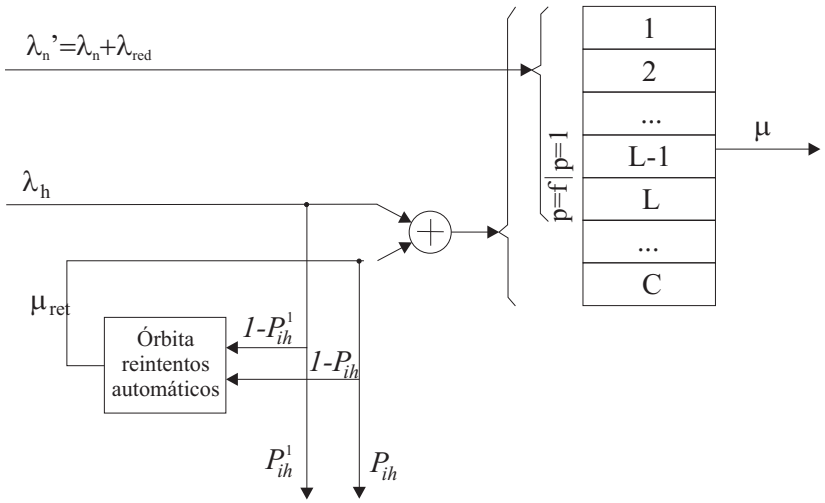


Figura 8.4: Modelo sin órbita de remarcados.

remarcados de forma que la tasa de llegadas se incrementa con un factor λ_{red}, cuyo valor dependerá de la solución empleada.

Los estudios realizados a este efecto utilizan la misma configuración de red que en la sección anterior, así nos encontramos con una situación de alta movilidad ($N_H = 2$). Es por ello que se ha decidido incluir también un caso en que se estudie brevemente lo que ocurre con una movilidad inferior ($N_H = 0.5$).

8.4.1 Escenario de movilidad alta

Modelo sin remarcados

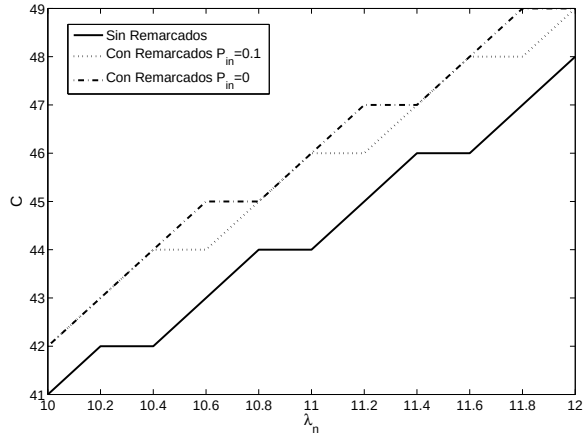
En este caso se ignora la existencia de remarcados así, en el modelo de la figura 8.4, se toma $\lambda_{red} = 0$. Esta solución provoca situaciones de bola de nieve bajo condiciones de sobrecarga producidas por los mismos remarcados tal y como se muestra en [TGM97]. Esto ocurre porque la red se ha diseñado con

el objetivo de obtener una determinada probabilidad de bloqueo y de terminación forzosa conforme a una tasa de entrada que no tiene en cuenta los reintentos. Sin embargo, cuando la red entra en funcionamiento los usuarios bloqueados remarcarán, de forma que el sistema estará sujeto a una carga adicional que no se había tenido en cuenta.

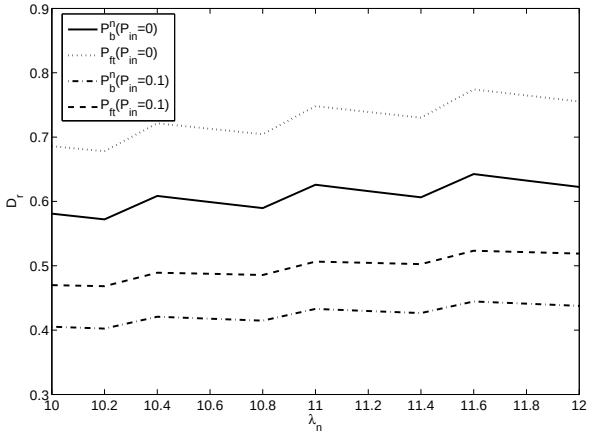
Para observar este efecto se han fijado unos objetivos de calidad servicio, en concreto, $P_b^n \leq 0.05$ y $P_{ft} \leq 0.005$ y se han diseñado dos redes para cumplir dichos objetivos. La primera red será una red sin remarcados —modelo de la figura 8.4 con $\lambda_{red} = 0$— y la segunda red será una red con remarcados —modelo de la figura 8.1—. En la figura 8.5 se observa el dimensionamiento de estos dos modelos para diferentes valores de λ_n —recuérdese que se ha tomado $\lambda_h = 2\lambda_n$—. Nótese que para la política de control de admisión *Fractional Guard Channel* se ha tomado $t = C - 1$ en todos los casos. Asimismo, cabe destacar que en el caso de que existan reintentos es necesario determinar la probabilidad de abandono, P_{in}, de los mismos. Se han considerado dos valores, $P_{in} = 0.1$ que puede ser un valor típico y $P_{in} = 0$ en cuyo caso no existe abandono. Como se puede observar en la figura 8.5(a) el número de recursos necesarios, C', para garantizar los objetivos de calidad de servicio en el sistema con remarcados es mayor que en el sistema sin remarcados. Esta diferencia se incrementa conforme disminuye el valor de P_{in}. En la figura 8.5(b) se muestra el incremento producido en la probabilidad de bloqueo de las sesiones nuevas, P_b^n, y en la probabilidad terminación forzosa, P_{ft}, si se utilizase el dimensionado del modelo sin remarcados en una red con remarcados. Es decir, se ha calculado C' para cumplir los requisitos de calidad de servicio en el modelo sin remarcados en distintas situaciones de carga, y se ha utilizado dicho C' para evaluar el modelo de la figura 8.1; calculando la diferencia relativa entre los parámetros de mérito obtenidos con los dos modelos para dicho valor C', definida como:

$$D_r = \frac{I(\text{modelo con remarcados}) - I(\text{modelo sin remarcados})}{I(\text{modelo sin remarcados})}$$

$$\text{con} \quad I = \{P_b^n, P_{ft}\}.$$



(a) C necesario para cumplir los requisitos de calidad de servicio



(b) Incremento en los parámetros de mérito al incluir los remarcados en un sistema diseñado sin considerarlos

Figura 8.5: Redimensionado para el modelo sin remarcados ($\lambda_{red} = 0$).

Naturalmente, para cada carga del sistema, λ_n, se ha calculado la C' correspondiente. Como se observa, aunque el incremento producido en el valor de los parámetros de mérito es mayor para $P_{in} = 0$ que para $P_{in} = 0.1$, en ambos casos es considerable.

Así se puede concluir que diseñar un sistema de comunicaciones sin considerar los remarcados producirá un infradimensionamiento de la red y, por tanto, una degradación en la calidad percibida por los usuarios.

Modelo Simplificado

Este modelo considera una carga adicional de peticiones nuevas que represente a la carga de remarcados y así, tener en cuenta la existencia de los remarcados. En este caso, en el modelo de la figura 8.4, a la tasa de llegadas de primeros intentos se le une una tasa adicional que represente la carga extra que suponen los reintentos y que vendrá dada por $\lambda_{red} = \mu_{red} N_{red}$, donde N_{red} es el número medio de usuarios en la órbita de remarcados del modelo de la figura 8.1. De este modo, los reintentos se consideran un primer intento más. Este modelo tiene un comportamiento totalmente opuesto al anterior, presentando problemas de sobredimensionamiento de la red. Supongamos que una petición es bloqueada y hasta que no realiza, por ejemplo, el quinto reintento no abandona el sistema definitivamente. El modelo simplificado considera, sin embargo, que han llegado seis peticiones y todas ellas han abandonado el sistema, cuando en realidad sólo se ha perdido una petición. Por lo tanto este modelo considera que hay más bloqueo del que en realidad existe y como consecuencia necesita aumentar el número de recursos del sistema para garantizar que se cumplen los objetivos de calidad de servicio.

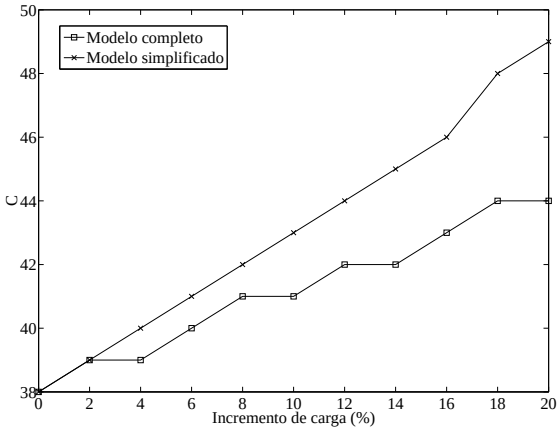
Para evaluar la magnitud del sobredimensionamiento se ha realizado el siguiente experimento. Se fijan unos objetivos de calidad de servicio ($P_b^n \leq 0.05$ y $P_{ft} \leq 0.005$) y se diseña la red para cumplir dichos objetivos. Partiendo de dicha situación se aumenta la tasa de llegada de peticiones nuevas, lo que produce un incremento de las probabilidades de bloqueo y de terminación forzosa. Este hecho, a su vez, requiere de un proceso de redimensionado con

el fin de reajustar el sistema para que vuelva a cumplir los objetivos de calidad de servicio. Para llevar a cabo dicho redimensionado se ha utilizado tanto el modelo con remarcados (Figura 8.1) como el modelo simplificado (Figura 8.4) donde los remarcados se consideran como una carga extra de peticiones nuevas. La figura 8.6 muestra los resultados que se obtienen conforme se incrementa la tasa de llegada de peticiones nuevas. El estudio se ha realizado para dos situaciones diferentes, cuando $P_{in} = 0$, es decir, cuando no hay impaciencia y para un valor $P_{in} = 0.1$ que es un valor típico para un sistema con impaciencia. Como se observa en ambos casos el redimensionado cuando se considera el modelo simplificado siempre requiere un mayor número de recursos que cuando se consideran los remarcados como tal (modelo de la figura 8.1). Obviamente para $P_{in} = 0$ la diferencia entre los dos modelos es superior.

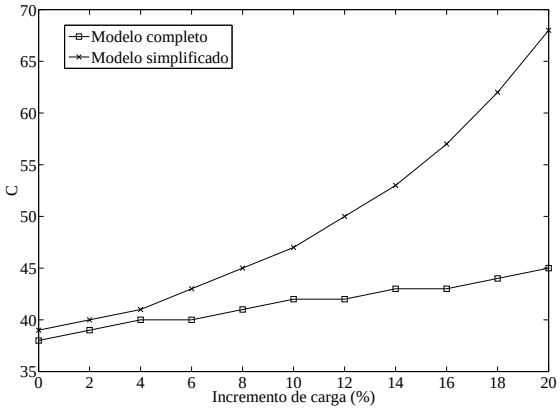
8.4.2 Escenario de movilidad baja

Igual que se ha hecho en la sección anterior, vamos a estudiar lo que ocurre cuando no se tienen en cuenta los remarcados —modelo sin remarcados— y cuando se consideran parte del flujo de sesiones nuevas —modelo simplificado— para un escenario con $N_H = \mu_{re}/\mu_s = 0.5$, $\mu = \mu_{re} + \mu_s = 1$. Al modificar la relación entre la tasa de servicio y la de residencia también cambiará la relación de tasas de llegada, considerando $\lambda_h = 0.5\lambda_n$. Por otro lado se ha considerado únicamente el caso sin impaciencia ($P_{in} = 0$). Asimismo se han fijado objetivos de calidad de servicio más estrictos, tomándose $P_b^n \leq 0.01$ y $P_{ft} \leq 0.001$.

De este modo, si observamos el efecto de dimensionar con le modelo sin remarcados, tal y como se hizo en la sección 8.4.1, se obtienen los resultados que se observan en la figura 8.7. Donde se observa un error en el dimensionado de la red similar al del caso anterior. El efecto de este subdimensionado será una degradación considerable de la calidad de servicio percibida por el usuario, con valores de probabilidad de bloqueo entre un 35 % y un 75 % por encima del objetivo de calidad establecido.



(a) $P_{in} = 0.1$.



(b) $P_{in} = 0$.

Figura 8.6: Redimensionado para el modelo simplificado ($\lambda_{red} = \mu_{red} N_{red}$)

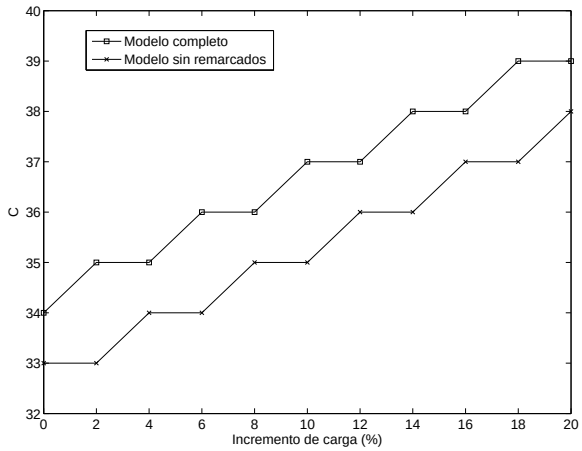


Figura 8.7: Mod. Sin remarcados: C necesario para cumplir objetivos.

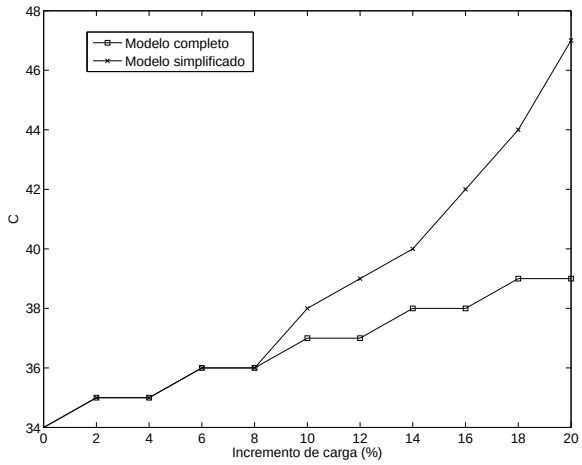


Figura 8.8: Mod. simplificado: C necesario para cumplir objetivos.

De la misma forma, si modelamos los remarcados como parte del flujo de sesiones nuevas —Modelo Simplificado— obtendremos un sistema sobredimensionado, tal y como se observa en la figura 8.8. Se puede observar que, aunque el error es menor que en el caso anterior, a partir de un incremento de la carga entorno al 10 % se produce un error de dimensionado apreciable y que puede ser importante en redes celulares debido a la escasez de recursos.

8.4.3 Conclusiones

Ignorar o tratar a los remarcados como parte del flujo de primeros intentos supone una degradación de la calidad de servicio percibida por el usuario o un uso ineficiente de los recursos debido al sobredimensionamiento, respectivamente. Obviamente, el primer el efecto es mucho más negativo que el segundo, pero en cualquier caso son efectos a evitar. Estos resultados ponen en evidencia la necesidad de considerar los remarcados como tal.

Estas mismas conclusiones se pueden trasladar al caso de los reintentos automáticos.

Capítulo 9

Sistemas de reintentos en la carecterización de aplicaciones

Si los sistemas de reintentos que hemos visto hasta el momento permiten modelar el comportamiento de los usuarios e, incluso, el de las propias redes de comunicaciones móviles, existen otro tipo de *reintentos* que permiten modelar el funcionamiento de las aplicaciones que se ejecutan sobre dichas redes de comunicaciones. En los sistemas de reintentos estudiados en los capítulos anteriores, tras un bloqueo, se reintenta tras un cierto tiempo con el fin de conseguir acceder a los recursos de la red. Sin embargo, para lograr dicho objetivo, existe otra posibilidad que ya no consiste en retardar el acceso a la red, sino en modificar las características de la comunicación que se quiere llevar a cabo, a otras menos exigentes. Así, si una sesión es bloqueada *reintentará* el acceso al sistema pero con unos requisitos menos restrictivos.

Este segundo tipo de reintento resultará muy útil a la hora de caracterizar las nuevas aplicaciones multimedia que se están implantando en las nuevas redes de comunicaciones y para las que se están desarrollando tecnologías de compresión y codificación que permiten que dichas aplicaciones sean adaptativas en tasa. Por ejemplo, una aplicación de video-telefonía genérica utiliza más de 40kbps, sin embargo para una video-telefonía de baja movilidad pue-

de ser suficiente con 25kbps. Con esta tasa se ofrece un calidad aceptable para este tipo de aplicación, tal y como se observa en [SJBD98]. Cuando estas aplicaciones piden una conexión a la red, el usuario especifica un rango discreto —o jerárquico— de tasas con el que puede trabajar la aplicación, definiendo una serie valores, desde el número mínimo de recursos ($MinBW$) hasta el máximo de recursos ($MaxBW$). Si la red tiene suficientes recursos disponibles, la petición se aceptará asignándole $MaxBW$, lo que significa que la sesión se acepta con alta calidad. Por el contrario, si la red está congestionada, será necesario disminuir el número de recursos asignados a la sesión. En caso de que se pueda garantizar la asignación de, al menos, $MinBW$ recursos a la sesión, ésta se aceptará, puesto que se aseguran unos requisitos mínimos de calidad. En caso de que no se pueda garantizar ni siquiera la asignación de $MinBW$, la sesión no se aceptará. Nótese que, aunque el ancho de banda asignado a la sesión cambia en función de la ocupación de la red, la duración de la sesión se mantiene fija. Este tipo de aplicación se denomina *rate adaptive*. Podemos encontrar algunos trabajos en este tipo de aplicación en [MVJ97, HJPS98], mientras que en [NWL97] podemos encontrar una variación de esta solución en que se realiza un cambio de codificador según las condiciones de la red.

A parte de las aplicaciones *rate adaptive* existen otro tipo de aplicaciones que también hacen uso del paradigma de tasa adaptativa a las que denominamos *elastic traffic* —o tráfico elástico—. Este segundo tipo se corresponde con aplicaciones relacionadas con la transferencia electrónica de documentos, como pueda ser la descarga de páginas web, envío de correos electrónicos, etc. En este caso, la adaptación de tasa es continua, de forma que la aplicación utilice cualquier ancho sobrante en la red. Sin embargo, la mayor diferencia con las *rate adaptive* radica en que la duración de la sesión sí se modifica con el ancho de banda. Es decir, a menor ancho de banda asignado a la sesión, mayor será el tiempo de duración de la misma. En estas aplicaciones el producto $tasa \times tiempo$ se mantiene constante [BR03]. Nótese que este tipo de aplicaciones no suele ser sensible al retardo o *jitter* que experimenten los paquetes como ocurre con la mayoría de aplicaciones *rate adaptive*, sino

al tiempo necesario para transmitir todo el documento. Esto hace que surjan nuevos parámetros de mérito como puede ser la probabilidad de abandono. El tiempo necesario para transmitir el documento depende del número de sesiones que comparten los mismos recursos. A mayor número de sesiones, el número de recursos por sesión disminuye y, como consecuencia, el tiempo de transmisión aumenta. Al aumentar el tiempo de transmisión es más probable que el usuario se impaciente y abandone el sistema sin terminar la transferencia. Estos abandonos resultarán de utilidad para tratar la sobrecargar, ya que ayudarán a estabilizar el sistema. Pero debe tenerse en cuenta que los abandonos impactan de forma muy negativa en las prestaciones del sistema puesto que el ancho de banda utilizado por las sesiones no terminadas se desperdicia. El paradigma de aplicaciones adaptativas juega un papel muy importante en las redes móviles celulares debido a la escasez de recursos y la importancia de hacer un uso eficiente de los mismos. Así, parece necesario que los mecanismos de adaptación de tasas colaboren con las políticas de control de admisión, con el fin de mejorar el rendimiento de estas redes.

El estudio de las aplicaciones adaptiva se introdujo originalmente en el estudio de redes fijas. Destacan los trabajos de Kaufman y Roberts [Kau81, Rob81] con la introducción del modelo *Erlang Multirate Loss Model* (EMLM) que, aunque sea un modelo para tráfico de *streaming*, es decir, sin adaptación de tasas, sirve como trampolín de salida al modelado de tráfico *rate adaptive* y *elastic traffic*. Este modelo permite calcular las probabilidades de bloqueo en un sistema en el que diferentes tipos de flujos, cada uno de ellos con diferentes requisitos de recursos y/o tiempos de servicios, compiten por los recursos disponibles bajo una política de *Complete Sharing* (CS) [HR86]. Puesto que se trata de tráfico de *streaming*, el número de recursos asignados es fijo, no pudiendo cambiar ni durante el establecimiento de la conexión ni durante el tiempo de servicio. El modelo EMLM presenta una estructura en forma-producto, lo cual asegura una alta precisión en el cálculo de las probabilidades de estado del sistema. Este modelo sirve como base para multitud de trabajos que estudian el caso de tráfico adaptativo [Kau92a, Kau92b, MLK02, SK97, RGF02]. En [Kau92a] se presenta el *retrial model* en

el que cuando un usuario es bloqueado, reintenta el acceso al sistema, pero cada vez pedirá menos recursos. Mientras que en este modelo se espera a que exista bloqueo para reducir el número de recursos pedidos, en el modelo propuesto en [Kau92b] se trata de evitar llegar a esa situación introduciendo umbrales. Así por ejemplo en el caso de considerar un único umbral, U_o, un flujo pedirá la asignación de unos recursos r en caso que la ocupación del sistema sea menor que U_o; en caso contrario pedirá una cantidad de recursos $r' < r$. En [MLK02] se extiende el modelo propuesto en [Kau92b] para el caso en cada clase de servicio tenga sus propios umbrales y no sean genéricos del sistema. En los modelos [Kau92a, Kau92b, MLK02] una vez se asignan recursos a una sesión estos ya no se modifican. Otras extensiones del modelo EMLM como [SK97, RGF02] sí que permiten modificar el ancho de banda de las sesiones en curso.

Todos estos modelos han sido desarrollados para redes fijas y, aunque se podría realizar una extensión de los mismos para su aplicación en redes móviles celulares se ha optado por otros modelos más complejos que introduzcan políticas de control de admisión. En la siguiente sección se estudia un escenario en el que se desarrolla un mecanismo de adaptación de tasas para funcionar junto con un mecanismo de control de admisión dinámico. El objetivo que se nos plantea es garantizar una calidad de servicio aceptable, a la vez que se limita la probabilidad de bloqueo experimentada por las diferentes clases de servicio y se asegura una alta utilización de los recursos de la red. Posteriormente se estudia un escenario en el que el mecanismo de control de admisión se modifica para garantizar los objetivos de calidad de servicio de un flujo de tráfico elástico.

9.1 RA: Política de adaptación de tasas

Con la proliferación de dispositivos inalámbricos como portátiles, PDAs o teléfonos móviles, la demanda de comunicaciones móviles ha crecido considerablemente en los últimos años y con ella, la necesidad de poder acceder a

las últimas aplicaciones multimedia. Teniendo en cuenta que muchas de estas aplicaciones multimedia son *rate adaptive* (RA), junto con la necesidad de hacer un uso eficiente de los escasos recursos, no es de extrañar la aparición de muchos trabajos relacionados con el estudio de políticas de adaptación de tasas en el entorno móvil.

La mayoría de estos trabajos consideran un escenario donde todos los recursos son compartidos, es decir, sin ningún mecanismo de control de admisión. Entre estos trabajos destaca [CPOG04], en el que los autores desarrollan, para un escenario multiservicio, una estrategia basada en la priorización de unos flujos respecto a otros, de forma que, en caso de congestión, los primeros flujos que vean disminuir su ancho de banda sean los menos prioritarios. Otro mecanismo de similares características se observa en [Sch05] donde el número de recursos asignado a cada sesión, en caso de congestión, es proporcional a la diferencia entre el valor máximo y el mínimo que se pueda asignar a dicho servicio. Los sistemas en que se comparten todos los recursos no son muy eficientes ya que, en general, sufren una tasa elevada de cambios que les hace inestables. Para evitar estos problemas algunos trabajos como [XCW02] introducen mecanismos de control de admisión. En el caso de [XCW02], se presenta un mecanismos de control de admisión basado en la medida de dos parámetros propios de las aplicaciones *rate adaptive*. En este mismo sentido en [CS02] se desarrolla una metodología analítica para determinar algunos de estos parámetros.

Otros trabajos han optado por mecanismos alternativos para determinar la política de adaptación de tasas. En [KCBN03] se propone un esquema de adaptación de tasas predictivo, en el que el control de admisión tiene en cuenta el estado de las células vecinas. Otra posibilidad para determinar la política de adaptación de tasas es utilizar un mecanismo de optimización de una determinada función de recompensa [AG08]. En [KCD02] se presenta un sistema para el cual se formulan dos problemas de optimización para maximizar recompensa y *fairness*.

En nuestro trabajo se ha optado por desarrollar un mecanismo de adap-

tación de tasas que trabaje junto con un esquema dinámico de control de admisión con el fin de garantizar una calidad de servicio aceptable a la vez que limita la probabilidad de bloqueo del sistema y asegura un uso eficiente de los recursos del sistema. Para poder explicar con detalle el mecanismo de adaptación de tasas empezaremos por comentar como funciona este mecanismo de control de admisión y luego introduciremos los cambios necesarios para que realice también la adaptación de tasas.

9.1.1 Esquema de control de admisión

El mecanismo de control de admisión que se ha utilizado es una modificación de la política *Multiple Guard Channel* (MGC) [CC97, LLC98] que permite modificar los umbrales definidos por dicha política de forma continua según el estado de la red. Se trata de un esquema dinámico ya que ajusta su configuración para cumplir los objetivos de calidad de servicio fijados, adaptándose así a cualquier configuración del tráfico entrante al sistema. Además el esquema utilizado permite asegurar un tratamiento diferenciado de las diferentes clases de servicio definidas en el sistema.

Consideremos el siguiente escenario en el que un conjunto de K clases de servicio compiten por los C recursos de los que dispone una célula. Se ha considerado el caso homogéneo en que todas las células son estadísticamente idénticas e independientes, lo que permite estudiar el comportamiento global del sistema a partir de una única célula. Nótese que el significado físico de una unidad de recursos depende de la implementación concreta que se haga del interfaz radio. Puesto que nos encontramos en un escenario celular, cada clase de servicio presentará dos tipos de llegadas, nuevas y *handover*, lo que da lugar a $2K$ flujos de llegada. Denotamos por s_i al i-ésimo flujo de llegada, con $1 \leq i \leq 2K$. Adicionalmente se denota como s_k^n (s_k^h), al flujo de llegada asociados con las peticiones nuevas (*handovers*) de la clase k, $1 \leq k \leq K$, siendo $s_k^n = s_k$ y $s_k^h = s_{k+K}$, $1 \leq k \leq K$.

Aunque el mecanismo de control de admisión desarrollado funciona para cualquier tipo de proceso de llegadas, así como para cualquier distribución

de tiempos de servicio y de residencia, por temas de tratabilidad matemática se ha optado por considerar que las peticiones de nuevas sesiones (*handovers*) de la clase de servicio k llegan al sistema según una distribución de Poisson con tasa λ_k^n (λ_k^h). Por otro lado, aunque no es necesario definir ninguna relación entre λ_k^n y λ_k^h, se ha supuesto, por simplicidad, que λ_k^h es una fracción constante de λ_k^n [Jab96, BS97]. Así, se denota como λ_{max} a la capacidad del sistema, es decir, la máxima λ que puede ofrecerse al sistema mientras que este es capaz de cumplir los objetivos de calidad de servicio. Nótese que λ es la tasa agregada de llegada peticiones de sesiones nuevas, es decir, $\lambda = \sum_{k=1}^{K} \lambda_k^n$, con $\lambda_k^n = f_k\lambda$ y $\sum_{k=1}^{K} f_k = 1$. El parámetro f_k define el factor de penetración de las diferentes clases de servicio. Es una aproximación muy común cuando se estudian este tipo de sistemas [BS97]. Asimismo, para una sesión de la clase k, tanto la distribución del tiempo de servicio como del tiempo de permanencia en la célula se supone que están distribuidos exponencialmente con tasas μ_k^s y μ_k^{re} respectivamente. De este modo, el *resource holding time* o tiempo de ocupación de los recursos también está exponencialmente distribuido con tasa $\mu_k = \mu_k^s + \mu_k^{re}$. La suposición de exponencialidad para el tiempo de permanencia en la célula representa una buena aproximación para este parámetro cuando lo que se está calculando son las probabilidades de bloqueo [KZ97]. Adicionalmente, una clase de servicio k, requerirá d_k unidades de recurso por sesión. Como cada clase de servicio está compuesta por dos flujos de llegada, si denominamos c_i a la cantidad de recursos que el flujo i necesita para cada sesión, entonces $d_k = c_k = c_{k+K}$, con $1 \leq k \leq K$. Si definimos $\boldsymbol{n} := (n_1, \ldots, n_K)$ como el vector de sesiones en curso, donde n_k es el número de sesiones en progreso de la clase k en la célula iniciadas, bien como sesiones nuevas, bien como *handovers*, entonces $c(\boldsymbol{n}) = \sum_{k=1}^{K} n_k d_k$ denota el número total de recursos ocupados en el estado $\boldsymbol{n}$.

Por otra parte es necesario definir las probabilidades de bloqueo de cada flujo ya que este es el parámetro de mérito que define la calidad de servicio que ofrece el sistema. Así denominamos P_{b_i}, con $1 \leq i \leq 2K$, a la probabilidad de bloqueo percibida por las peticiones del flujo s_i. Y de igual modo que hemos hecho anteriormente con los flujos, se utilizarán las definiciones

$P^n_{b_k} = P_{b_k}$ ($P^h_{b_k} = P_{b_{K+k}}$) para definir la probabilidad de bloqueo percibida por las peticiones de sesiones nuevas (*handovers*) de la clase k. Nótese que los objetivos de calidad de servicio vienen dados como límite superior a las probabilidades de bloqueo experimentadas por cada flujo de llegada. Para diferenciar el bloqueo experimentado de los objetivos, se define B^k_n (B^k_h) como el límite para la probabilidad de bloqueo de sesiones nuevas (*handovers*) de la clase k.

El esquema de control de admisión utilizado básicamente se encarga de ajustar los parámetros de configuración de la política MGC, al estado de la red. Esta política define un parámetro de configuración por cada flujo de entrada i, $l_i \in \mathbb{N}$. A este parámetro se le suele denominar umbral o *threshold*. Cuando llega una petición de servicio de un flujo de llegada i en el estado $\boldsymbol{n}$ se acepta si $c(\boldsymbol{n}) + c_i \leq l_i$, en caso contrario se bloquea. Por lo tanto, l_i es la cantidad de recursos del sistema a los que tiene acceso el flujo i, de forma que aumentar (disminuir) su valor supone una reducción (aumento) de la P_{b_i}.

La mayoría de esquemas de control de admisión dinámicos propuestos están basados en la reserva de recursos y, por lo tanto, en el uso de *guard channels*. Este es el caso, por ejemplo, de los trabajos [ZL01, WZZZ03] para escenarios monoservicio. Este tipo de políticas permite a los operadores conseguir una alta utilización de los recursos tal y como se demuestra en [GMP05]. Muchos de estos trabajos, [ZL01, WZZZ03, YL97, RSAK99, YK02], utilizan ventanas temporales de medida para estimar los parámetros del sistema. La utilización de estas ventanas temporales tiene como consecuencia, la necesidad de un tiempo de convergencia demasiado largo para afrontar condiciones reales de funcionamiento, o bien una baja precisión. Para evitar estos problemas el esquema utilizado realiza un ajuste probabilístico de los parámetros de configuración de MGC cada vez que se toma una decisión de admisión, ya sea de aceptación o de rechazo, evitando así el uso de ventanas temporales de medida. De este modo el esquema de ajuste desarrollado se encuentra siempre en el bucle de la figura 9.1, adaptando los umbrales del sistema de forma continua. Nótese que este bucle afecta únicamente a un flujo y, puesto que se van a estudiar sistemas multiservicio, vamos a necesitar

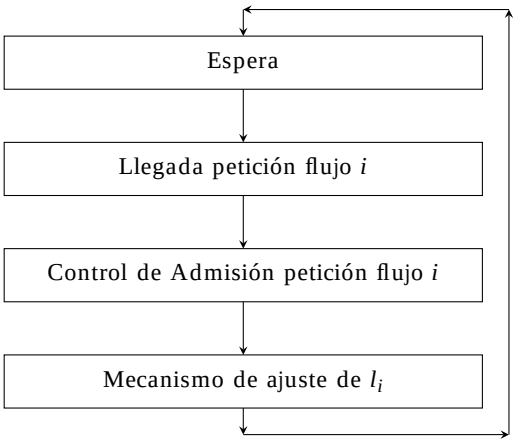


Figura 9.1: Esquema del control de admisión dinámico

un esquema como el de la figura para cada flujo.

El funcionamiento del esquema se define, por motivos de simplicidad, para el caso en que los procesos de llegada son estacionarios y el sistema se encuentra en régimen permanente. Partimos de la siguiente definición del objetivo de bloqueo para un flujo s_i, $B_i = b_i / o_i$, donde $b_i, o_i \in \mathbb{N}$. El esquema tratará de ajustar el parámetro l_i de forma que $P_{b_i} = B_i$, es decir, se busca que las peticiones de servicio del flujo i experimenten, en media, b_i rechazos y $o_i - b_i$ aceptaciones sobre un total de o_i peticiones ofrecidas. Por ejemplo, para un objetivo de $B_i = 1/100$ se deben experimentar $b_i = 1$ pérdidas cada $o_i = 100$ peticiones. Partiendo de esta idea, parece intuitivo pensar que el esquema no debe cambiar el valor de los umbrales de aquellos flujos que cumplen sus objetivos de calidad de servicio. Mientras que se debe ajustar el valor de l_i, en el sentido adecuado, en el caso que el flujo i no cumpla sus objetivos. Es decir, se debe aumentar el valor del umbral si la probabilidad de bloqueo experimentada supera el objetivo establecido para ese flujo, y disminuir su valor en caso contrario. Así se propone el siguiente ajuste probabilístico:

- Si se acepta la petición: $\{l_i = (l_i - 1)\}$ con probabilidad $1/(o_i - b_i)$;
- Si se rechaza la petición: $\{l_i = (l_i + 1)\}$ con probabilidad $1/b_i$.

Con este tipo de ajuste se consigue, que en condiciones de tráfico estacionario, cuando $P_{b_i} = B_i$, l_i, se asegura la igualdad $P_{b_i}\frac{1}{b_i} = (1 - P_{b_i})\frac{1}{o_i - b_i}$. Nótese además que, cuando el tráfico es no-estacionario, este esquema está continuamente ajustando la configuración de la política MGC con el fin de ajustarse a los objetivos de calidad de servicio el mayor tiempo posible.

De este modo, conseguimos ajustar el umbral del flujo s_i a partir de las decisiones de aceptación y rechazo del mismo flujo, de forma independiente a lo que ocurre en el resto de flujos. Sin embargo, cuando es necesario aumentar el valor del umbral del flujo s_i —nos encontramos en una situación de congestión— puede darse el caso de que $l_i \geq C$, lo que no ofrece beneficios adicionales y el sistema se ve abocado a una situación en que no es capaz de garantizar el objetivo de calidad de servicio del flujo s_i. Para evitar esta situación, el ajuste probabilístico diseñado incorpora la posibilidad de que un proceso de ajuste interactúe con los procesos de ajuste del resto de flujos. Para conseguir dicha interacción es necesario, primero, clasificar las diferentes flujos en dos categorías: i) aquellos para los que se definen objetivos de calidad de servicio y a los que llamaremos *protegidos*; ii) y flujo *Best-Effort Class* (BEC), para el que no se definen objetivos de calidad de servicio y por tanto experimenta una calidad de servicio impredecible. Dentro de los flujos protegidos, el operador define un orden de prioridad con el fin de proteger a los flujos de mayor importancia, ya sea por el tipo de aplicaciones que hagan uso de esos flujos, o por el coste económico de la contratación de los mismos. Se supone que los índices asociados a cada flujo corresponden con su prioridad, por lo tanto $\mathbf{s} = (s_1, s_2, \ldots, s_{2K})$ define el orden de prioridad del sistema, donde s_1 será el flujo de mayor prioridad (*Highest-Priority Class*, HPC) y s_{2K} el de menor prioridad (*Lowest-Priority Class*, LPC). Obviamente, si existe flujo BEC, éste será el LPC.

Dado un orden de prioridad, si el flujo s_i se encuentra en una situación en que $l_i \geq C$, el sistema pondrá en marcha un modo indirecto de ajuste

que ayuda al flujo s_i reduciendo el acceso a recursos de los flujos con menor prioridad. Para ello se disminuye el umbral l_j, siendo $j > i$. La forma de llevar a cabo este proceso es secuencial, es decir, se empieza disminuyendo el umbral del flujo de menor prioridad, s_{2K}, y, en caso de que este llegue a cero, se pasa a disminuir el umbral del siguiente flujo de menor prioridad, s_{2K-1}. Obviamente cuando un flujo s_i necesita ajustar el umbral de otro flujo de menor prioridad, s_j, el esquema de ajuste debe modificar l_j únicamente cuando lleguen peticiones del flujo s_i y no cuando lleguen peticiones del flujo s_j. Cuando esto ocurre, se dice que el esquema de ajuste asociado a s_j se ha desactivado. Nótese que un flujo j permanecerá desactivado mientras uno o varios de los flujos con mayor prioridad que j se encuentren congestionados, es decir presenten $l_i > C$.

Así existen dos modos de tratar la congestión, un modo directo en que se aumenta el umbral del flujo que experimenta la congestión, y un modo indirecto en que se disminuye el umbral de los flujos con una prioridad menor que la del flujo que experimenta la congestión. En el apéndice C.5 se muestran los diagramas de funcionamiento del esquema de ajuste probabilístico.

En general, el sistema se puede dimensionar como una CTMC multidimensional, donde el vector de estados viene dado por $(n_1, \ldots, n_K, l_1, \ldots, l_{2K})$. Recordemos que n_k es el número de sesiones en curso de la clase de servicio k, ya sean nuevas o *handovers*, mientras que $l_i \in \mathbb{N}$ es el umbral de aceptación asociado al flujo i. Nótese que se permite que l_i tome valores positivos y negativos con el fin de recordar los ajustes realizados así como para identificar el tipo de ajuste que se está utilizando, directo o indirecto. Dado que dibujar el diagrama general es complicado, se ha tomado como ejemplo el caso bidimensional que se muestra en la figura 9.2. Este sistema presenta una única clase de servicio y por tanto dos flujos, s^h y s^n, con $d = 1$ y C recursos en la célula. Asimismo, se considera que s^h es el flujo más prioritario, mientras que el s^n será un flujo *Best-effort*. Además, por motivos de simplicidad, las peticiones del flujo s^h se aceptan siempre que existan recursos disponibles en el sistema. Así, su umbral se utilizará únicamente como indicador de congestión en el sistema y, por tanto, para limitar el acceso a los recursos del flujo

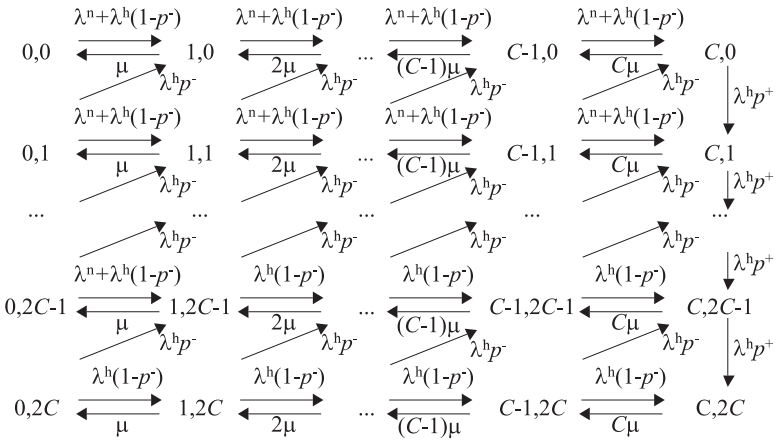


Figura 9.2: Diagrama de transiciones del control de admisión dinámico.

s^n. Con esta configuración, el vector de estados se puede definir como (n, l^h), donde n es el número de unidades de recurso ocupadas. Como s^n es un flujo *Best-Effort* el único parámetro a ajustar es el umbral del flujo s^h. Este umbral, l^h, se ajusta siguiendo el ajuste probabilístico descrito anteriormente. Para ello, en la figura 9.2 aparecen dos nuevos parámetros, p^- y p^+, que definen las probabilidades de cambiar el valor de l^h. Así, si el objetivo de calidad de servicio de los *handovers* viene dado por $B^h = \frac{b^h}{o^h}$, donde b^h representa las peticiones bloqueadas por cada o^h peticiones ofrecidas, entonces $p^- = \frac{1}{o^h - b^h}$ y $p^+ = \frac{1}{o^h}$. Por otro lado, para controlar el acceso al sistema de las peticiones del flujo s^n se recurre a la norma siguiente $l^n = C - \max\{0, (l^h - C)\}$. Nótese que mientras nos encontremos en una situación de carga baja tendremos que $l^n = C$, mientras que en casos de sobrecarga el ajuste del flujo s^h pasará al modo indirecto de ajuste, en cuyo caso l^n se verá reducido conforme a la sobrecarga experimentada por el flujo s^h.

La tabla 9.1 muestra las tasas de transición del sistema descrito, nótese que aunque el vector de estados puede definirse como (n, l^h), en esta tabla se ha optado por una descripción más detalla utilizando la tupla (n^n, n^h, l^n, l^h).

En dicha tabla aparecen las funciones $\alpha_n(\mathbf{x})$, $\alpha_h(\mathbf{x})$ y $\beta(\mathbf{x})$, que se definen

Tabla 9.1: Tasas de transición. Estado actual $\mathbf{x} = (n^n, n^h, l^n, l^h)$.

Estado siguiente	Tasa de transición
(n^n+1, n^h, l^n, l^h)	$\lambda^n \cdot \alpha_n(\mathbf{x})$
(n^n, n^h+1, l^n, l^h)	$\lambda^h \cdot (1-p_h^-) \cdot \alpha_h(\mathbf{x})$
$(n^n, n^h+1, l^n, l^h-\Delta l)$	$\lambda^h \cdot p_h^- \cdot \alpha_h(\mathbf{x}) \cdot \beta(\mathbf{x})$
$(n^n, n^h+1, l^n+\Delta l, l^h-\Delta l)$	$\lambda^h \cdot p_h^- \cdot \alpha_h(\mathbf{x}) \cdot (1-\beta(\mathbf{x}))$
$(n^n, n^h, l^n, l^h+\Delta l)$	$\lambda^h \cdot p_h^+ \cdot (1-\alpha_h(\mathbf{x})) \cdot \beta(\mathbf{x})$
$(n^n, n^h, l^n-\Delta l, l^h+\Delta l)$	$\lambda^h \cdot p_h^+ \cdot (1-\alpha_h(\mathbf{x})) \cdot (1-\beta(\mathbf{x}))$
(n^n-1, n^h, l^n, l^h)	$n^n\mu$
(n^n, n^h-1, l^n, l^h)	$n^h\mu$

como:

$$\alpha_n(\mathbf{x}) = \begin{cases} 1 & (n^n + n^h < C) \cap (n^n + n^h < l^n) \\ 0 & (n^n + n^h = C) \cup (n^n + n^h \geq l^n) \end{cases}$$

$$\alpha_h(\mathbf{x}) = \begin{cases} 1 & (n^n + n^h < C) \\ 0 & (n^n + n^h = C) \end{cases}$$

$$\beta(\mathbf{x}) = \begin{cases} 1 & l^h \leq C \\ 0 & l^h > C \end{cases}$$

Para evaluar este mecanismo de ajuste probabilístico se ha realizado un estudio para el siguiente escenario:

C	d_1	d_2	f_1	f_2	λ_r^n	λ_r^h	μ_1	μ_2	$B_1^n(\%)$	$B_2^n(\%)$	$B_r^h(\%)$
50	2	4	0.8	0.2	$f_r\lambda$	$0.5\lambda_r^n$	1	3	5	1	$0.1B_r^n$

Modificándose el valor de λ con el fin de estudiar el comportamiento del sistema en diferentes situaciones de carga. Asimismo se considera el siguiente orden de prioridad $\{s_2^h, s_1^h, s_2^n, s_1^n\}$, siendo s_2^h el flujo más prioritario y s_1^n el menos prioritario. Nótese que se considera que todos los flujos tienen requisitos de calidad de servicio, no incluyéndose ningún flujo *Best-Effort*.

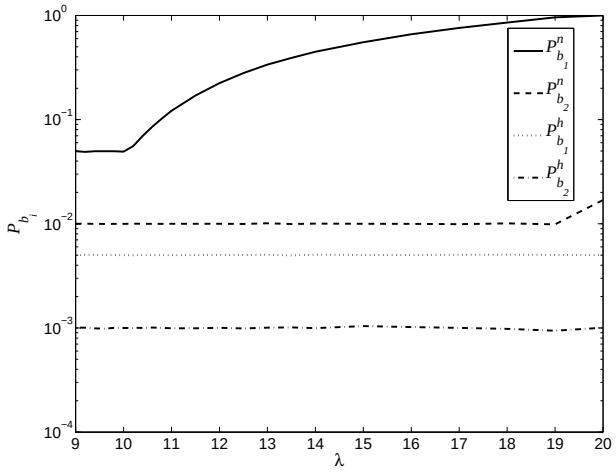


Figura 9.3: P_{b_i} cuando utilizamos el control de admisión dinámico.

La figura 9.3 muestra el valor de la probabilidad de bloqueo de los cuatro flujos del escenario, P_{b_i} conforme varia la carga del sistema. Nótese como el esquema de ajuste trata de asegurar que $P_{b_i} = B_i$ siempre que sea posible, obteniéndose una carga máxima —entendida como el máximo valor para el cual se cumplen los objetivos de calidad de servicio— de $\lambda_{Max} = 10.3$. Como se observa en la figura 9.3 , el flujo menos prioritario, s_1^n, es el primer penalizado al aumentar la carga, así será el primero en dejar de cumplir su objetivo de probabilidad de bloqueo. En el caso de que la congestión siga aumentando y el umbral de este flujo llegue a cero, y por tanto, la probabilidad de bloqueo del mismo alcance el 100 %, será necesario penalizar al siguiente flujo menos prioritario, s_2^n.

Este mecanismo ofrece muy buenos resultados, consiguiendo ajustar la configuración de la política MGC con una excelente precisión. Además presenta un tiempo de convergencia corto y sin oscilaciones. En [GR07] se realiza una evaluación muy completa de este mecanismo.

9.1.2 Inclusión de la política de adaptación de tasas

La política de adaptación de tasas, a la que denominamos RA, aprovecha la adaptabilidad de las aplicaciones multimedia para limitar la probabilidad de bloqueo tanto de sesiones nuevas como de *handovers* durante los periodos de congestión. El funcionamiento de la política RA propuesta difiere considerablemente de otras soluciones existentes. La mayoría de estas soluciones considera una compartición total de recursos donde, en caso de congestión, la tasa de las sesiones en curso disminuye con cada nueva sesión o *handover* que llega al sistema y se incrementa cada vez que sale una sesión. La política RA propuesta hace uso del mecanismo de detección anticipada de congestión que provee el sistema de umbrales del control de admisión para determinar la adaptación de tasas. Asimismo, se ha decidido no modificar la tasa de las sesiones en curso. Se ha optado por esta solución porque, aunque modificar las tasas de las sesiones en curso suele llevar a una mejor utilización de los recursos del sistema, supone una variación en la calidad percibida por el usuario, para el cual resulta más negativo recibir un servicio de calidad variable, que un servicio con una calidad constante aunque baja [Gir92, XCW02, AG08].

El funcionamiento de la política RA está basado en los mismos principios descritos para el control de admisión dinámico. En este caso, mientras no exista congestión, es decir, mientras todos los umbrales se encuentren por debajo de C, sólo se ejecutarán los mecanismos de ajuste de umbrales del control de admisión. Cuando algún flujo entre en congestión, $l_i > C$, se pondrá en marcha el mecanismo de adaptación de tasas. Si el control de admisión lo que hacía era limitar el acceso a recursos de los flujos menos prioritarios. Ahora, con la inclusión de la política RA, se optará por disminuir la tasa binaria asignada a los flujos de menor, igual y superior prioridad, en este orden. Nótese que la congestión en un flujo dado puede afectar a la tasa no sólo de los flujos de menor prioridad sino también a los de mayor prioridad. El orden de prioridad es definido libremente por el operador de red. En este caso, por simplicidad se considera el mismo orden que el definido para el control de admisión. Para el usuario es más desagradable perder una peti-

ción de sesión nueva o de *handover* que obtener un servicio de menor tasa. Aunque esta disminución de tasa disminuye la calidad percibida, se asegura que aún se ofrezca una calidad aceptable. En caso de que, tras disminuir la tasa de todos los flujos el sistema, todavía se encuentre en una situación de congestión, se activará el mecanismo indirecto definido para el esquema de control de admisión. Esto garantiza que por lo menos los flujos de mayor prioridad cumplan sus objetivos de calidad de servicio.

Pongamos un ejemplo del funcionamiento de la política RA utilizando el escenario del apartado anterior al que se le añade la posibilidad de que las sesiones de ambas clases de servicio puedan disminuir su tasa a la mitad. El estado de la política RA viene definido por dos matrices, la matriz de tasas y la matriz de probabilidad de utilización de dichas tasas. La matriz de tasas indica las diferentes tasas que puede utilizar un flujo y que vendrán dadas por $c_{i,j}$ con $i \leq 2K$ y $j \leq N$, con N el número de valores diferentes que se puede dar a la tasa de un flujo cualquiera. Para el caso del escenario considerado tenemos que:

$$\mathbf{c} = \begin{bmatrix} c_{2h,1} & c_{2h,2} \\ c_{1h,1} & c_{1h,2} \\ c_{2n,1} & c_{2n,2} \\ c_{1n,1} & c_{1n,2} \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 1 \\ 4 & 2 \\ 2 & 1 \end{bmatrix}. \tag{9.1}$$

De este modo, por ejemplo, al flujo s_1^n —el menos prioritario—, se le asignarán $c_{1n,1} = 2$ recursos en caso de que no exista congestión en el sistema, mientras que cuando aparezca congestión su tasa se verá reducida a $c_{1n,2} = 1$ recursos. Nótese que en el caso de que aparezca congestión en el sistema, el primer flujo en ver reducida su tasa será el menos prioritario pero, si con esto no es suficiente, se pasará a disminuir la tasa de los flujos con una prioridad inmediatamente superior. Para poder llevar a cabo este proceso de la forma más suave posible y al mismo tiempo evitar que los flujos sufran una mayor degradación de la necesaria, se define una matriz de probabilidad de utilización asociada a la matriz de tasas, que vendrá dada por $p_{i,j}$ con $i \leq 2K$ y $j < N$. Con la definición de estas dos matrices, podemos decir que el número

medio de recursos, $E[c_i]$, asignados a una sesión del flujo s_i es $E[c_i] = c_i \times p_i^t$, o dicho de otro modo $E[c_i] = c_{i,1}p_{i,1} + c_{i,2}p_{i,2} + \ldots + c_{i,N}p_{i,N}$. El objetivo del esquema de RA es disminuir este valor medio conforme aumente la congestión en el sistema. Originalmente la configuración de esta matriz será:

$$\mathbf{p} = \begin{bmatrix} p_{2h,1} & p_{2h,2} \\ p_{1h,1} & p_{1h,2} \\ p_{2n,1} & p_{2n,2} \\ p_{1n,1} & p_{1n,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \tag{9.2}$$

de forma que todos los flujos estén utilizando su tasa —asignación de recursos— máxima. Cuando aparezca congestión se empezará disminuyendo la tasa del flujo menos prioritario, para ello se disminuirá $p_{1n,1}$ y se aumentará $p_{1n,2}$, pero siempre garantizándose que $p_{1n,1} + p_{1n,2} = 1$. En caso de que se llegue a la situación en que $p_{1n,2} = 1$ y la congestión perdure, será necesario disminuir la tasa asignada al flujo de prioridad inmediatamente superior, es decir, s_2^n.

Veamos la evolución de la matriz $\mathbf{p}$ para el escenario estudiado conforme aumenta la cogestión. En este caso se ha decido tomar como probabilidades de utilización para todos los flujos $p_{i,j} \in \{1, 0.5\}$. De este modo, partiendo de la situación mostrada en la matriz (9.2), conforme aumenta la congestión $\mathbf{p}$ pasará por los siguientes estados:

$$\mathbf{p} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0.5 & 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \rightarrow \ldots$$

Como se observa, inicialmente todos los flujos piden la tasa máxima, $c_{i1}\ \forall i$; cuando aumenta la congestión se pasa a disminuir la tasa del menos prioritario, es decir el flujo $s_4 = s_1^n$, para ello se aceptan la mitad de las peticiones con tasa máxima, $c_{1n,1} = 2$ y la otra mitad con tasa mínima $c_{1n,2} = 1$. Si con esto no es suficiente, se pasa a aceptar todas las peticiones del flujo s_1^n con la tasa mínima, $c_{1n,2} = 1$. Si la congestión sigue aumentando, el siguiente paso para asegurar el cumplimiento de los requisitos de calidad de servicio es disminuir la tasa del siguiente flujo de menor prioridad, s_2^n, aceptando la mitad de

sus peticiones con tasa máxima, $c_{2n,1}$ y la otra mitad con tasa mínima, $c_{2n,2}$, y así sucesivamente. La evolución de la política RA depende de la evolución de la congestión. Conforme aumenta la congestión de los flujos que acceden al sistema, estos son degradados con mayor severidad. Nótese que si se llega a:

$$\mathbf{p} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix},$$

ya no será posible disminuir la tasa de ningún flujo, pues todas están al mínimo. En este caso, si la congestión perdura, será necesario recurrir al modo indirecto de actuación del control de admisión dinámico que vimos en la sección anterior.

Para evaluar esta política vamos ha realizar el mismo experimento que se hizo para el control de admisión dinámico. Así se variará el valor de λ con el fin de estudiar el comportamiento del sistema para diferentes puntos de congestión. En ausencia de política RA este sistema, tal y como vimos en el apartado anterior, es capaz de soportar una carga máxima de $\lambda_{Max} = 10.3$. Así, si se define la carga relativa como $(\lambda - \lambda_{Max})/\lambda_{Max}$, los valores de carga relativa utilizados para realizar la evaluación se encuentran entre -0.1 y 2.5.

Las figuras 9.4, 9.5 muestran los resultados obtenidos al aplicar la política RA. En concreto, la figura 9.4 muestra la variación de las probabilidades de bloqueo de los diferentes flujos al modificar la carga del sistema. Nótese que, al introducir la política RA, el sistema soporta una tasa de llegadas mayor a la que soportaría el mismo sistema sin la política RA. Así, el tráfico máximo soportado por este sistema sin superar ninguno de los objetivos de calidad de servicio es de 32.96, para una $\lambda = 23.69$ —es decir una carga relativa de 1.3—, frente a los 27.75 para una $\lambda = 10.3$ que soportaba el sistema sin la política RA. Esta mejora se debe a la posibilidad de disminuir la tasa de las sesiones entrantes tan pronto como se detecte congestión, lo que permite aceptar más peticiones de servicio. Nótese además que, cuando la tasa de llegadas supera

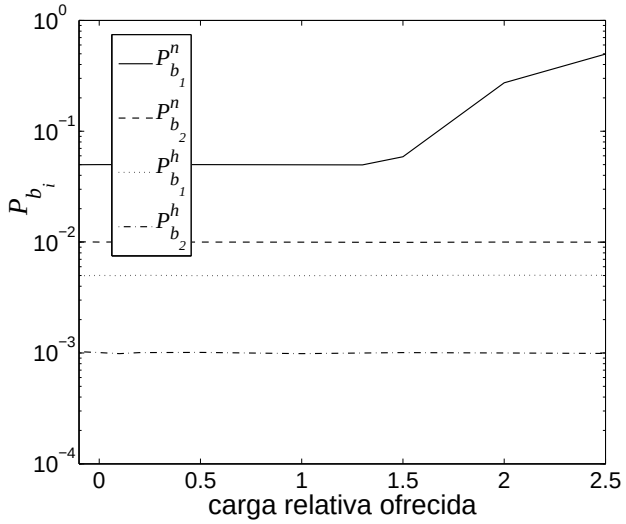


Figura 9.4: Variación de P_{bi} cuando se aplica la política RA.

$\lambda = 23.69$, el esquema de control de admisión pone en marcha el modo indirecto de ajuste de umbrales y el objetivo de bloqueo definido para el flujo de menor prioridad ya no se cumple.

En la figura 9.5 se observa el porcentaje de tráfico que se cursa degradado (curvas con marcador, en concreto un diamante) y sin degradación (curvas sin marcador) conforme varía la tasa de sesiones ofrecida al sistema. Cuando el sistema experimenta congestión el flujo menos prioritario, s^n_1, es el primero que degrada su tasa. Si la congestión persiste el resto de flujos también se ven obligados a reducir sus tasas en orden inverso al orden de prioridad definido.

Desde un punto de vista operacional, la política RA funciona de forma independiente en cada célula, y por tanto sólo tiene una visibilidad parcial de la vida de las sesiones. El modelo propuesto permite ajustar el funcionamiento de la política RA al funcionamiento de unidades de gestión que mantengan información sobre la evolución de las sesiones. Esto será de gran utilidad en escenarios de alta movilidad donde la tasa de *handovers* sea alta, como puede

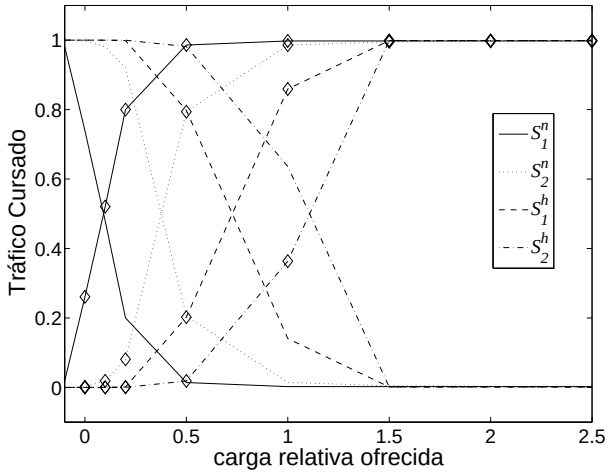


Figura 9.5: Trafico degradado y no-degradado cuando se aplica RA.

ser el caso de áreas geográficas servidas por picocélulas. En ese sentido el mecanismo de RA se puede considerar como una entidad que ofrece *consejos* de adaptación de tasas que pueden ser seguidos por el sistema o no. Debe tenerse en cuenta que, no seguir estos *consejos*, hará que la política RA evolucione hacia situaciones de mayor severidad pudiendo, incluso, afectar a los flujos de mayor prioridad o llegar a disparar la activación del modo indirecto del control de admisión. En conclusión, el esquema RA diseñado ofrece una gran flexibilidad a la hora de configurar los servicios móviles.

9.2 Mecanismo de Control de Admisión para tráfico elástico

Otra situación que nos podemos encontrar en las redes actuales es la existencia, junto con el tráfico de *streaming*, de tráfico elástico correspondiente a aplicaciones basadas en la transferencia de documentos electrónicos. El trá-

fico elástico presenta unos requisitos temporales bajos y puede adaptarse a los recursos disponibles. Es por ello que se de prioridad al tráfico de tiempo real y se ofrezca al tráfico elástico la capacidad sobrantes —se puede reservar un pequeño porcentaje de los recursos a este tipo de tráfico para prevenir la inanición de este tráfico en condiciones de sobrecarga—. Generalmente se utiliza TCP para transportar este tipo de tráfico, protocolo que se encargará de la adaptación de tasas y de la compartición del ancho de banda entre los diferentes flujos. Conforme el ancho de banda asignado a cada sesión se reduce, la probabilidad de que un usuario abandone el sistema sin haber completado el servicio debido a la impaciencia, aumenta. Esta impaciencia, debida a un bajo *throughput*, puede aparecer debida a la impaciencia humana o a que TCP o protocolos de capas superiores interpretan que la conexión se ha roto. Los abandonos son útiles para tratar la sobrecarga puesto que ayudan a estabilizar el sistema. Sin embargo, este fenómeno también tiene un impacto negativo en la eficiencia de la red. Este hecho llevó a los autores de [BR03] a reivindicar la necesidad de utilizar mecanismos de control de admisión para los flujos de tráfico elástico.

Puesto que nuestro estudio se centra en el interfaz radio de la red de acceso, cada flujo elástico estará limitado en tasa, bien debido a las características del terminal móvil, bien debido al cuello de botella que supone el enlace radio. De este modo, cada flujo recibe un ancho de banda determinado, siempre inferior a un determinado valor que será común para todos los terminales. Se considera, por tratabilidad matemática, que la cantidad de información a transmitir (dada en *bytes*) está exponencialmente distribuida. Aunque se acepta que la distribución estadística del tamaño de los documentos en Internet presenta una variabilidad superior a la de una distribución exponencial, los resultados obtenidos en [BR03] demuestran que los resultados obtenidos con una distribución exponencial pueden ser considerados como un límite inferior para el rendimiento del sistema.

Consideramos el mismo escenario que el descrito en 9.1.1, en el que tenemos flujos de tiempo real, al que se añade un flujo *best effort* de tráfico elástico. Se denota como s_e^n (s_e^h), el flujo de llegadas asociado a las peticiones

de sesiones nuevas (*handovers*) del flujo elástico. Dichas peticiones llegan según un proceso de Poisson con una tasa variable en el tiempo $\lambda_e^n(t)$ ($\lambda_e^h(t)$). Para una sesión elástica se considera que el tiempo de residencia en la célula se encuentra exponencialmente distribuido con tasa μ_e^{re}. Si se denota como d_e el número máximo de unidades de recursos que puede pedir una sesión elástica, y como n_e el número de sesiones elásticas en el sistema, entonces la tasa de servicio se define como:

$$\mu_e^s = \begin{cases} \mu_e^s & \text{cuando } n_e d_e \leq (C - c(\boldsymbol{n})) \\ \mu_e^s (C - c(\boldsymbol{n}))/(n_e d_e) & \text{cuando } n_e d_e > (C - c(\boldsymbol{n})) \end{cases}$$

donde $c(\boldsymbol{n})$ es el número de unidades de recursos ocupados por los flujos de tiempo de real. Para modelar completamente el comportamiento de las sesiones elásticas es necesario considerar la impaciencia. Para ello se define el tiempo de impaciencia como una variable aleatoria independiente y distribuida exponencialmente. La tasa de impaciencia, μ_I, se asume como inversamente proporcional a la cantidad de recursos que se asignan a cada sesión elástica:

$$\mu_I = \begin{cases} 0 & \text{cuando } n_e d_e \leq (C - c(\boldsymbol{n})) \\ W(n_e d_e/(C - c(\boldsymbol{n}))) & \text{cuando } n_e d_e > (C - c(\boldsymbol{n})) \end{cases}$$

donde W es una constante. Asimismo, se denota como BA el objetivo de calidad de servicio fijado para los flujos elásticos. Este objetivo viene dado en términos de la probabilidad de abandono, fijándose un valor máximo para la misma. Esta probabilidad de abandono define el porcentaje entre sesiones no completas y sesiones aceptadas. Por otro lado, se define como PA a la probabilidad de abandono que realmente experimenta el flujo elástico.

Para que este tipo de tráfico elástico sea capaz de cumplir sus requisitos de calidad de servicio, ofreciendo una probabilidad de abandono inferior a la objetivo el esquema de control de admisión definido en 9.1.1 debe ser modificado. Para ello, se define un parámetro de configuración, $l_e \in \mathbb{N}$, que se asociara al flujo s_e^n, de forma similar a los umbrales definidos para los flujos de tiempo real. Cuando hay n_e sesiones elásticas en curso, una petición de servicio por parte de una nueva sesión será aceptada si $n_e + 1 \leq l_e$ y

bloqueada en caso contrario. Las peticiones de servicio por parte de *handovers* se aceptan siempre, independientemente de la ocupación del sistema. Para definir el comportamiento del esquema de gestión de umbrales utilizaremos una aproximación similar a la que se describió para el esquema de ajuste del control de admisión dinámico. Así, el objetivo de un flujo elástico se expresa como $BA = a/b$, donde a es el número de sesiones no completadas y b el número de sesiones aceptadas en el sistema, con $a, b \in \mathbb{N}$. Partiendo de este objetivo se define el siguiente ajuste probabilístico del umbral l_e:

- $\{l_e = (l_e - 1)\}$ con probabilidad $1/a$ cada vez que una sesión elástica abandona debido a impaciencia.
- $\{l_e = (l_e + 1)\}$ con probabilidad $1/(b-a)$ cada vez que una sesión elástica completa si servicio de forma satisfactoria, es decir termina la sesión o realiza un *handover* hacia otra célula.

Se ha evaluado, por simulación, las prestaciones de este esquema para un escenario con tráfico de tiempo real y elástico. En concreto, se ha estudiado un escenario con las siguientes características:

C	d_1	d_2	f_1	f_2	λ_r^n	λ_r^h	μ_1	μ_2	$B_1^n(\%)$	$B_2^n(\%)$	$B_r^h(\%)$
10	1	2	0.8	0.2	$f_r\lambda$	$0.5\lambda_r^n$	1	3	5	1	$0.1B_r^n$

Se considera que el tráfico de tiempo real ofrece una carga constante e igual a la capacidad del sistema ($\lambda = \lambda_{max} = 1.89$). Para evitar la inanición del tráfico elástico se reserva 1 unidad de recursos para este tráfico. El resto de parámetros que definen el tráfico elástico son: $\mu_e^s = 2.0$, $\mu_e^d = 2.0$, $K = 0.4$, $\lambda_e^h = 0.5\lambda_e^n$, con un objetivo de calidad de servicio de $BA = 0.1$.

La figura 9.6 muestra como el esquema desarrollado asegura el cumplimiento del objetivo de calidad de servicio. Si no se implementa dicho esquema la probabilidad de abandono aumenta, conforme la tasa de tráfico elástico

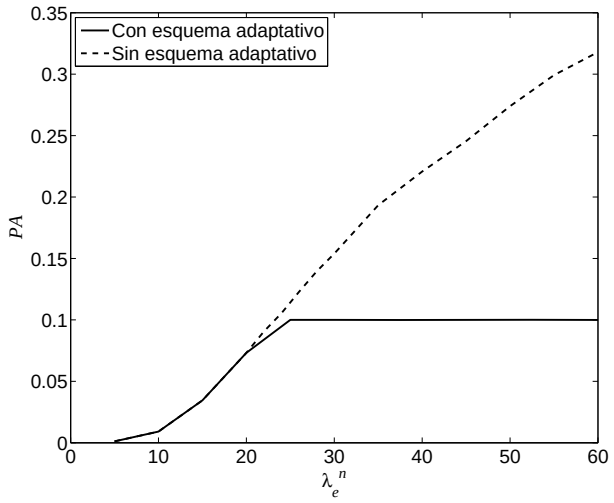


Figura 9.6: Probabilidad de abandono del tráfico elástico

aumenta. Esto se debe al hecho de que, al aumentar la tasa de sesiones elásticas que llegan al sistema, a cada sesión elástica se le asignan menos recursos puesto que se están aceptando más sesiones. La reducción del número de recursos asignados aumenta el tiempo de servicio y como consecuencia la tasa de impaciencia. Nótese además que una tasa alta de abandonos tiene como consecuencia un uso ineficiente de los recursos del sistema puesto que los recursos que se asignan a sesiones que no llegan a completarse se desperdician totalmente.

9.3 Conclusiones

Se ha desarrollado una política de reserva de recursos capaz de trabajar con tráfico *rate adaptive* así como con tráfico elástico. Los resultados de los diferentes estudios realizados muestran que se consigue una degradación suave de las prestaciones de los diferentes flujos *rate adaptive* cuando existe congestión

en el sistema, lo que permite aumentar tanto el tráfico cursado, como la tasa de llegada de usuarios que es capaz de soportar el sistema. Adicionalmente, se ha visto como se puede incluir un flujo de tráfico elástico como *best effort* con el fin de aprovechar los recursos que los flujos de tiempo real dejan libres sin que esto afecte a la calidad de servicio obtenida por los flujos de tiempo real.

Capítulo 10

Conclusiones

La gestión de recursos en las redes móviles celulares se presenta como un mecanismo de gran importancia para este tipo de redes que han crecido considerablemente en los últimos años y en para las que están apareciendo nuevos tipos de aplicaciones con unos requisitos de ancho de banda mayores.

En este trabajo se han abordado dos aspectos a tener en cuenta por los diferentes mecanismos de gestión de recursos implementados. Se ha estudiado en profundidad el efecto que el reintento, tanto el debido al comportamiento humano como el debido a las características de este tipo de red, tiene en las prestaciones del sistema. En este sentido, los capítulos 4 y 5 se han desarrollado diferentes aproximaciones, basadas en la resolución de las ecuaciones de Kolmogorov, para modelar este comportamiento, estudiando la repuesta de cada una de ellas en un amplio abanico de escenarios. Del estudio de estas evaluaciones, en el capítulo 6 se concluye que los modelos Infinitos presentan las mejores prestaciones tanto en términos de complejidad del espacio de estados a resolver como en coste computacional, destacando principalmente el modelo HM2 que presenta un buen compromiso entre complejidad y coste computacional. Por otra parte, en el capítulo 7 se ha presentado la posibilidad de recurrir a otro tipo de solución, denominada *Value Extrapolation*, que ya no está basada en las ecuaciones de Kolmogorov, sino que define el pro-

blema como un *Markov Decision Process* que se puede resolver haciendo uso de las ecuaciones de Howard. Este método permite, por sus características, adaptarse a escenarios más complejos.

Mientras que en los capítulos anteriores todos los estudios se han realizado para un escenario genérico típico en el estudio de sistemas de reintento, en el capítulo 8 se ha introducido este efecto en el modelo de una red celular. La complejidad y características del modelo resultante en el que aparecen dos dimensiones infinitas y heterogéneas han hecho imposible la aplicación de los modelos desarrollados en el capítulo 5 a ambas órbitas de reintentos. Aunque otras soluciones son posibles, se ha recurrido a la aplicación del modelo FM a cada una de las órbitas con el fin de tener un espacio de estados finito en todas las dimensiones y así poder resolver el sistema. En este sentido, al tener que modelar una red celular, ha sido necesario introducir algunas características propias de estos sistemas. Así se ha introducido un mecanismo de control de admisión basado en la reserva de recursos para priorizar los *handovers* sobre las sesiones nuevas. Además se ha diseñado un mecanismo que consiguiera el comportamiento determinista, propio de los reintentos automáticos de los *handovers*, a partir de una distribución exponencial, más sencilla de implementar. El modelo de la red celular desarrollado nos ha permitido demostrar que el efecto de los reintentos en las redes móviles celulares no es despreciable y que ignorar su existencia puede tener graves consecuencias en el rendimiento de la red. Así, se puede llegar a situaciones de fuerte degradación de la calidad de servicio experimentada por los usuarios. Mientras que considerar los reintentos como parte de los flujos de primeros intentos llevará a un sobredimensionamiento de la red y, por tanto, a un uso poco eficiente de los recursos.

Observando los resultados de los diferentes modelos propuestos junto con los de aquellos que se han propuesto en la literatura parece obvio que recurrir a modelos como el FM no es la mejor solución. Sin embargo, por las características del escenario resulta imposible utilizar un modelo infinito. Se podría aplicar un modelo infinito a una de las órbitas y uno finito a la otra, pero esto no mejoraría sustancialmente los resultados. Así se plantea

como trabajo futuro el desarrollo del modelo *Value Extrapolation* definido en el capítulo 7 para el caso de dos órbitas. Este modelo no recurre al cálculo del vector de probabilidades de estado y por tanto evita el problema derivado de las dos dimensiones infinitas y heterogéneas y además, como se ha visto en el capítulo 7, obtiene muy buenos resultados, especialmente en términos de coste computacional.

Por otra parte, se ha estudiado la forma de aprovechar las características de las nuevas aplicaciones multimedia que se vienen implantando en los últimos años en las redes celulares. Estas aplicaciones presentan la posibilidad de de modificar sus requisitos de recursos según el estado de la red. De este modo, en el capítulo 9 se han desarrollado diferentes esquemas para trabajar junto con un mecanismo dinámico de control de admisión. El objetivo de este trabajo era mejorar el uso de los recursos de los que dispone el sistema, sin perder de vista la necesidad de cumplir los objetivos de calidad de servicio. Por simplicidad, se han estudiado de forma separada los esquemas necesarios para tratar tráfico *rate adaptive* y elástico, y se plantea, como trabajo futuro, unir estos esquemas en una única solución que incluya todas las posibilidades. Los resultados de la política RA desarrollada muestran que se puede conseguir una degradación suave de las prestaciones de los diferentes flujos *rate adaptive*, lo que permite aumentar el tráfico cursado. Por otra parte, se ha desarrollado un mecanismo que asegura una probabilidad de abandono baja para el tráfico elástico. Este mecanismo es necesario para evitar un mal uso de los recursos de la red. Los resultados obtenidos del estudio del tráfico elástico nos ha permitido observar como, en escenarios con tráfico de tiempo real, la inclusión de este tipo de tráfico como un flujo *best effort* permite aprovechar los recursos que los flujos de tiempo real dejan libres.

Además de las diversas tareas apuntadas que completarían el trabajo realizado, se pueden estimar, a grandes rasgos, las líneas de investigación que se consideran más relevantes, vista la evolución que están siguiendo las redes de acceso móvil. Al analizar el desarrollo de las redes de acceso móvil destaca la fuerte introducción de aplicaciones multimedia. Entre estas aplicaciones encontramos tanto aplicaciones de tiempo real como de transferencia

de datos, todas ellas con una gran demanda de ancho de banda y la capacidad de adaptar sus demandas al estado de la red. Aunque en el capítulo 9 se han desarrollado algunos mecanismos orientados a la gestión de este tipo de aplicaciones, estos solo pueden entenderse como un punto de partida desde el cual profundizar en un área que puede ser de vital importancia en la gestión de las futuras redes de comunicaciones. Además, con el desarrollo de multitud de tecnologías inalámbricas, en las futuras redes de acceso móvil coexistirán diferentes tecnologías de acceso. En ese sentido la integración de diferentes tecnologías de acceso inalámbricas en un mismo terminal se está convirtiendo en una realidad. Este cambio de escenario llevará asociado consigo la necesidad de realizar una gestión eficiente de forma que el usuario este *always best connected*.

Apéndices

Apéndice A

Abreviaturas y acrónimos

AP	Método de Artalejo y Pozo propuesto en [AP02]
BEC	*Best Effort Class*
CS	*Complete Sharing*
CSMA / CD	*Carrier Sense Multiple Access with Collision Detection*
CTMC	*Continuous Time Markov Chain*
EMLM	*Erlang Multirate Loss Model*
Fal	Modelo de Falin propuesto en [Fal83]
FGC	*Fractional Guard Channel*
FM	Modelo Finito propuesto
FR	Modelo de Fredericks y Reisner propuesto en [FR79]
GJL	Algoritmo de Gaver, Jacobson y Latouche propeusto en [GJL84]
GPRS	*General Packet Radio Service*
GSM	*Groupe Spécial Mobile / Global System for Mobile Communications*
GW	Modelo de Greenwerg y Wolff propuesto en [GW87]
HM1,HM2	Modelos infinitos de homogeneización propuestos
HPC	*Highest Priority Class*
Int	Modelo deInterpolación
IPP	Proceso Interrumpido de Poisson
LDQBD	*Level-dependet QBD*

LM	Modelo infinito de limitación del espacio de estados propuesto
LPC	*Lowest Priority Class*
MAP	*Markovian Arrival Process*
Mar	Modelo de Marsan et al propuesto en [MCL+01]
MDP	Proceso de Decisión de Markov
MGC	*Multiple Guard Channel*
NR	Modelo de Neuts y Rao propuesto en [NR90]
QBD	*Quasi Birth and Death Process*
RA	*Rate Adaptive*
RTA	*Returning Customers See Time Average*
UMTS	Universal Mobile Telecommunications Service
VE	*Value Extrapolation*

Apéndice B

Notación, variables y parámetros más utilizados

Escenario Monoservicio

C	número de unidades de recurso del sistema
U	Tamaño de la población
λ	tasa de llegadas
λ_n	tasa de llegadas de peticiones de sesiones nuevas
λ_h	tasa de llegadas de peticiones de *handover*
$1/\mu$	tiempo medio de ocupación de los recursos
$1/\mu_s$	tiempo medio de duración de la sesión
$1/\mu_{re}$	tiempo medio de residencia en una célula
μ_r	tasa de reintentos (genérica)
μ_{red}	tasa de remarcados
μ_{ret}	tasa de reintentos automáticos
P_i	probabilidad de impaciencia de los reintentos (genérica)
P_{in}	probabilidad de impaciencia de los remarcados
P_{ih}	probabilidad de impaciencia de los reintentos automáticos
P_{in}^1	probabilidad de impaciencia del 1º intento de una petición de sesión nueva
P_{ih}^1	probabilidad de impaciencia del 1º intento de una petición de *handover*

t	parámetro de configuración de la política *Fractional Guard Channel*
$1/\mu_r'$	tiempo de residencia en el área de solape
$\mathbf{Q}$	Generador infinitesimal
$\mathbf{A}_i^{(j)}$	Submatriz del generador infinitesimal
$\pi(k,m)$	probabilidad de estado
$\mathbf{R}$	*rate matrix*
Q	nivel de truncación de los modelos aproximados
Q_n	nivel de truncación de la órbita de remarcados
Q_h	nivel de truncación de la órbita de reintentos automáticos
P_b^n	probabilidad de bloqueo de una sesión nueva
P_b^h	probabilidad de bloqueo de *handover*
P_{ft}	probabilidad de terminación forzosa
P_{si}	probabilidad de servicio inmediato
P_{sd}	probabilidad de servicio demorado
P_{ns}	Probabilidad de no servicio
c_s	coste / recompensa de realizar una acción en el estado s
g	coste / recompensa promediada
$q_{ss'}$	tasa de transición de un estado s a otro s'
w_s	*relative state value*
Escenario Multiservicio	
K	número de clases de servicio diferentes
k	servicio ($1 \leq k \leq K$).
i	flujo de peticiones ($1 \leq i \leq 2K$).
c_k	número de unidades de recurso para cursar una sesión de la clase k
d_i	número de unidades de recurso para cursar una sesión del flujo i
λ	tasa agregada de llegada peticiones de sesiones nuevas
f_k	penetración del servicio
λ_k^n	tasa de llegada de peticiones de establecimiento de nueva sesión
λ_k^h	tasa de llegadas de peticiones de *handover*
$\lambda_{\text{máx}}$	capacidad del sistema
n_k	número de sesiones en progreso de un servicio
$c(\boldsymbol{n})$	número de unidades de recurso ocupadas del sistema

$1/\mu_i$	tiempo medio de ocupación de los recursos
$1/\mu_k^s$	tiempo medio de duración de la sesión
$1/\mu_k^{re}$	tiempo medio de residencia en una célula
$P_{b_k}^n$	probabilidad de bloqueo de una sesión nueva del servicio k
$P_{b_k}^h$	probabilidad de bloqueo de un *handover* del servicio k
B_k^n	cota superior objetivo para $P_{b_k}^n$
B_k^h	cota superior objetivo para $P_{b_k}^h$
l_k^n l_k^h	parámetros de configuración de la política *Multiple Guard Channel*
d_{kM_k}	tasas que el servicio k puede utilizar
ϕ_{im_i}	tasa a la que operan las peticiones del flujo s_i
p_{in_i}	probabilidad de degradación de cada flujo
$\lambda_e^n(t)$	tasa de llegadas de peticiones de establecimiento de nueva sesión elástica
$\lambda_e^h(t)$	tasa de llegadas de peticiones de *handovers* elásticos
$1/\mu_e^s$	tiempo medio de duración de la sesión de una sesión elástica
$1/\mu_e^{re}$	tiempo medio de residencia en una célula de una sesión elástica
d_e	número de unidades de recurso para cursar una petición elástica
n_e	número de sesiones elásticas en progreso
$1/\mu_I$	tiempo medio de impaciencia de las sesiones elásticas
l_e	parámetros de configuración (umbrales) para un flujo elástico
BA	Probabilidad de abandono objetivo de un flujo elástico
PA	Probabilidad de abandono experimentada por un flujo elástico

Apéndice C

Expresiones matemáticas y esquemas de funcionamiento

C.1 Método de Bright y Taylor

Para poder resolver *QBDs* no-homogéneos (LDQBD) e infinitos es necesario evaluar la familia de matrices $\mathbf{R}_k$ con $k \geq 0$, que son las soluciones mínimas no negativas del sistema de ecuaciones

$$\mathbf{A}_0^{(k)} + \mathbf{R}_k[\mathbf{R}_{k+1}\mathbf{A}_2^{(k+2)}] = 0 \quad k \geq 0.$$

Por otro lado, Ramaswami en [Ram95] muestra que la distribución de equilibrio $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots)$ para un LDQBD recurrente positivo satisface la relación

$$\mathbf{x}_{k+1} = \mathbf{x}_k\mathbf{R}_k \quad k \geq 0,$$

donde $\mathbf{x}_0$ es la solución a

$$\mathbf{x}_0[\mathbf{A}_1^{(0)} + \mathbf{R}_0\mathbf{A}_2^{(1)}] = 0, \tag{C.1}$$

sujeta a la condición de normalización.

Para poder calcular dicha distribución de equilibrio es necesario conocer $\mathbf{R}_k$ y, en general, solo es posible calcularlas de forma numérica. Así, será necesario truncar el espacio de estados, definiéndose $(\mathbf{x}_k(K))_j$, $0 \leq k \leq K$ como la probabilidad en régimen permanente de que la cadena $X(t)$ se encuentre en el estado (k, j), condicionado a que nos encontremos dentro de la truncación. Con ellos podemos expresar las probabilidades de estado como:

$$\mathbf{x}_k(K) = \mathbf{x}_0(K) \prod_{m=0}^{k-1} \mathbf{R}_m \tag{C.2}$$

donde $\mathbf{x_0}$ debe satisfacer la ecuación (C.1) así como

$$\mathbf{x}_0(K) \sum_{k=0}^{K} [\prod_{m=0}^{k-1} \mathbf{R}_m]\mathbf{e} = 1. \tag{C.3}$$

Bright y Taylor en [BT97] presentan una forma de calcular la distribución en equilibrio para el modelo truncado mediante una aproximación que permite considerar el efecto de los estados que quedan fuera de la truncación. Este hecho, como veremos, permite hacer referencia a este método como solución exacta, cuando en realidad se está realizando una aproximación.

Este método se basa, dada una truncación K, en el cálculo de las matrices $\mathbf{R}_k$ en $0 \leq k \leq K-1$. A partir de $\mathbf{R}_{K-1}$ se calcularán el resto de matrices $\mathbf{R}_k$ mediante el proceso iterativo

$$\mathbf{R}_k = \mathbf{A}_0^{(k)}(-\mathbf{A}_1^{(k+1)} - \mathbf{R}_{k+1}\mathbf{A}_2^{(k+2)})^{-1}.$$

Obtenidas las matrices $\{\mathbf{R}_k\}$, se pueden obtener los vectores de probabilidades de estado en régimen permanente, empezando con $\mathbf{x}_0$:

$$\mathbf{x}_0(K)(\mathbf{A}_1^{(0)} + \mathbf{R}_0\mathbf{A}_2^{(1)}) = 0,$$

junto con la condición de normalización, $\mathbf{x}_0(K)\mathbf{e} = 1$. Se obtendrán el resto de vectores a partir de $\mathbf{x}_0$ mediante el proceso iterativo

$$\mathbf{x}_k(K) = \mathbf{x}_{k-1}(K)\mathbf{R}_{k-1},$$

donde será necesario normalizar cada uno de los vectores.

La parte más compleja, por tanto, radica en el cómputo de $\mathbf{R_{K-1}}$. Para obtener esta matriz se define el siguiente algoritmo

1: $l = 0$
2: $\mathbf{U} = \mathbf{U}_k^0$, $\mathbf{D} = \mathbf{D}_{k+2}^0$
3: $\mathbf{P} = \mathbf{I}$
4: Inicializamos $\mathbf{R}_{K-1}(0) = \mathbf{U}$
5: Iterar:
6: $\quad l = l + 1$
7: $\quad \mathbf{P} = \mathbf{D} * \mathbf{P}$
8: $\quad \mathbf{U} = \mathbf{U}_k^l$, $\mathbf{D} = \mathbf{D}_{k+2^{l+1}}^l$
9: $\quad \mathbf{R}_k(l) = \mathbf{R}_k(l-1) + \mathbf{U} * \mathbf{P}$
10: hasta que $(\mathbf{R}_k(l) - \mathbf{R}_k(l-1))_{max} < \epsilon$
11: $\mathbf{R}_k = \mathbf{R}_k(l)$ con $k = K - 1$.

Donde los valores de las matrices $\mathbf{U}$ y $\mathbf{D}$ vendrán dados por

$$\mathbf{U}_k^0 = \mathbf{A}_0^{(k)}(-\mathbf{A}_1^{(k+1)})^{-1}$$
$$\mathbf{D}_{k+2}^0 = \mathbf{A}_2^{(k+2)}(-\mathbf{A}_1^{(k+1)})^{-1}$$
$$\mathbf{U}_k^l = \mathbf{U}_k^{l-1}\mathbf{U}_{k+2^{l-1}}^{l-1}[\mathbf{I} - \mathbf{U}_{k+2^l}^{l-1}\mathbf{D}_{k+3*2^{l-1}}^{l-1} - \mathbf{D}_{k+2^l}^{l-1}\mathbf{U}_{k+2^{l-1}}^{l-1}]^{-1}$$
$$\mathbf{D}_{k+2^{l+1}}^l = \mathbf{D}_{k+2^{l+1}}^{l-1}\mathbf{D}_{k+3*2^{l-1}}^{l-1}[\mathbf{I} - \mathbf{U}_{k+2^l}^{l-1}\mathbf{D}_{k+3*2^{l-1}}^{l-1} - \mathbf{D}_{k+2^l}^{l-1}\mathbf{U}_{k+2^{l-1}}^{l-1}]^{-1}$$

Con esto quedaría resuelto el sistema truncado. Faltaría por decidir un valor de K apropiado para realizar la truncación. En este sentido se busca un valor que haga que la probabilidad de estar en el último nivel de la truncación sea muy bajo, y por tanto despreciable.

Como se puede observar es un proceso complejo que tendrá un coste alto de resolución. Los sistemas de reintentos que tratamos de resolver entran en la categoría de *QBDs* no-homogéneos e infinitos, es por ello que solo podríamos recurrir a este método de Bright y Taylor para su resolución. Debido a su alto coste computacional, en general, se ha descartado el uso de métodos matemáticos y se ha tendido al uso de modelos aproximados que permitan una resolución más sencilla y rápida. No obstante hay que tener en cuenta

que los modelos aproximados que se plantean para resolver este tipo de sistemas van a generar otros procesos *QBD* más sencillos que los que aparecen en el modelo original y por tanto, habrá que recurrir a algunos de los métodos matemáticos que hemos visto para resolver *QBDs level-independent* o por lo menos finitos.

C.2 Métodos matemáticos para la resolución de QBDs

Son varios los métodos probados para la resolución de *QBDs*. En nuestro caso se implementan dos algoritmos, el propuesto por Gaver et al [GJL84] y el propuesto por Servi [Ser02], para la resolución de un sistema de reintentos de población finita. Estos dos algoritmos se han utilizado tanto para la resolución exacta del sistema como para la resolución del modelo FM que lo aproxima.

Aquí se desarrollan los pasos a seguir para resolver el caso exacto. Los cambios a realizar en los algoritmos para implementar la aproximación que realiza el modelo FM son directos a partir de la resolución exacta.

C.2.1 Algoritmo Propuesto por Gaver et Al [GJL84].

Para el caso del modelo exacto, el algoritmo consta de los siguientes pasos:

Paso 1) Cálculo de las matrices $\overline{\mathbf{C}}_k$ de tamaño $(C+1)\times(C+1)$ mediante la recursión inversa, siguiente:

$$\overline{\mathbf{C}}_k = \mathbf{S}_1^{(k)} + \mathbf{A}_0^{(k)}(-\overline{\mathbf{C}}_{k+1}^{-1})\mathbf{A}_2^{(k+1)};$$

Para $k = \{M-C-1,..,1,0\}$ y con la condición inicial $\overline{\mathbf{C}}_{M-C} = \mathbf{A}_1^{(M-C)}$

Paso 2) Evaluar el vector de probabilidades π_0:

$$\pi_0\overline{\mathbf{C}}_0 = \mathbf{0} \tag{C.4}$$

donde $\boldsymbol{\pi_0}$ y $\mathbf{0}$ son vectores fila de tamaño $(C+1)$.

Paso 3) Cálculo del resto de vectores de probabilidad mediante recursión directa:

$$\pi_k = \pi_{k-1}\mathbf{L}_{k-1}(-\overline{\mathbf{C}}_k^{-1}); k = 1, 2, ..., M - C. \tag{C.5}$$

Normalizando el resultado:

$$\sum_{k=0}^{M-C} \pi_k \mathbf{e} = 1 \tag{C.6}$$

C.2.2 Algoritmo Propuesto por Servi [Ser02]

En este algoritmo se parte de la definición de un vector de probabilidades de estado, de forma que el vector $\boldsymbol{e}_j = (e_{j0}, \ldots, e_{jn})^T$ representa la probabilidad de encontrarnos en los estados $(j,0), \ldots, (j,n)$. A partir de esta definición, el generador infinitesimal del sistema vendrá dado por tres tipos de submatrices, las matrices $[v_j^-]_{i,k}$ que definen las transiciones desde (j,i) hasta $(j-1,k)$, las matrices $[v_j^+]_{i,k}$ que definen las transiciones desde (j,i) hasta $(j+1,k)$ y las matrices $[v_j^0]_{i,k}$ que definen las transiciones desde (j,i) hasta (j,k), donde los elementos de la diagonal $(i = k)$ vendrán dados de forma que la suma de cada fila de la matriz $\mathbf{Q}$ sea cero. Según esta definición de las matrices, tendremos una nueva estructura en el diagrama de estados que se corresponde a la observada en la figura C.1.

Con este diagrama de estados, la matriz $\mathbf{Q}$ tendrá el siguiente aspecto:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{v}_0^0 & \mathbf{v}_0^+ & 0 & \ldots & 0 & 0 & 0 \\ \mathbf{v}_1^- & \mathbf{v}_1^0 & \mathbf{v}_1^+ & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \mathbf{v}_{C-1}^- & \mathbf{v}_{C-1}^0 & \mathbf{v}_{C-1}^+ \\ 0 & 0 & 0 & \ldots & 0 & \mathbf{v}_C^- & \mathbf{v}_C^0 \end{bmatrix}$$

Donde cada submatriz está definida del siguiente modo:

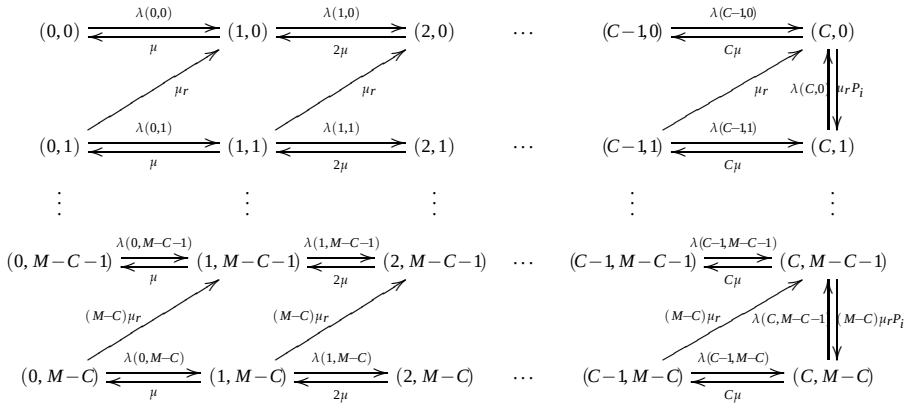


Figura C.1: Diagrama de transiciones para el algoritmo de Servi.

$$\mathbf{v}_j^0 = \begin{bmatrix} * & 0 & 0 & \dots & 0 & 0 \\ 0 & * & 0 & \dots & 0 & 0 \\ 0 & 0 & * & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * & 0 \\ 0 & 0 & 0 & \dots & 0 & * \end{bmatrix} \text{ para } j = 0, 1, 2, .., C-1$$

$$\mathbf{v}_C^0 = \begin{bmatrix} * & \lambda_C & 0 & \dots & 0 & 0 \\ \mu_r P_i & * & \lambda_{C+1} & \dots & 0 & 0 \\ 0 & 2\mu_r P_i & * & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * & \lambda_{M-1} \\ 0 & 0 & 0 & \dots & (M-C)\mu_r P_i & * \end{bmatrix}$$

$$\mathbf{v}_j^- = \begin{bmatrix} j\mu & 0 & 0 & \dots & 0 & 0 \\ 0 & j\mu & 0 & \dots & 0 & 0 \\ 0 & 0 & j\mu & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & j\mu & 0 \\ 0 & 0 & 0 & \dots & 0 & j\mu \end{bmatrix} \text{ para } j = 1, 2, .., C$$

$$\mathbf{v}_j^+ = \begin{bmatrix} \lambda_j & 0 & 0 & \dots & 0 & 0 \\ \mu_r & \lambda_{j+1} & 0 & \dots & 0 & 0 \\ 0 & 2\mu_r & \lambda_{j+2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_{j+M-C-1} & 0 \\ 0 & 0 & 0 & \dots & (M-C)\mu_r & \lambda_{j+M-C} \end{bmatrix} \text{ para } j = 0, 1, 2, .., C-1$$

La solución ha de satisfacer

$$\chi_{\{j\neq 0\}}\mathbf{e}_{j-1}^T v_{j-1}^+ + \mathbf{e}_j^T v_j^0 + \chi_{\{j\neq m\}}\mathbf{e}_{j+1}^T v_{j+1}^- = \mathbf{0}^T, \quad j = 0, \dots, m. \tag{C.7}$$

donde

$$\chi_{\{j\neq i\}} = \begin{cases} 0 & \text{si } j = i \\ 1 & \text{si } j \neq i \end{cases}$$

Y además se ha de verificar la condición de normalización:

$$\sum_{j=0}^{m} \mathbf{e}_j^T \mathbf{1} = 1 \tag{C.8}$$

Las ecuaciones (C.7) y (C.8) contienen una ecuación redundante. Por lo tanto, sin pérdida de generalidad podemos cambiar para el caso $j = m$ la última columna de la ecuación (C.7) por la ecuación $e_{mn} = 1$. Esto implica un cambio en las submatrices que forman **Q** del siguiente modo:

$$[v_m^0]_{ik} = \begin{cases} [v_m^0]_{ik} & \text{para } k \neq n \\ \\ \chi_{\{i=n\}} & \text{para } k = n \end{cases} \quad y \quad [v_{m-1}^+]_{ik} = \chi_{\{k\neq n\}}[v_{m-1}^+]_{ik}$$

Y reemplazar en la ecuación (C.7) el $\mathbf{0}^T$ por $(0,\ldots,0,\chi_{\{j=m\}})$.

En [Ser02] se proponen dos algoritmos diferentes para resolver procesos *QBD* definidos de este modo.

Algoritmo 0

1. Hacer $\hat{v}_0^0 = v_0^0$.
2. Para $j = 1,\ldots,m \;\rightarrow\; \hat{v}_j^0 = v_j^0 - v_j^-(\hat{v}_{j-1}^0)^{-1}v_{j-1}^+$.
3. $\mathbf{e}_m^T = (0,\ldots,0,1)(\hat{v}_m^0)^{-1}$.
4. Para $j = m-1,\ldots,0 \;\rightarrow\; \mathbf{e}_j^T = -\mathbf{e}_{j+1}^T v_{j+1}^-(\hat{v}_j^0)^{-1}$
5. Normalización $\rightarrow\; \mathbf{e}_j^T = \frac{\mathbf{e}_j^T}{\sum_{j=0}^m \mathbf{e}_j^T\mathbf{1}}$

Algoritmo 1

1. Se define $\mathbf{E}_0 = \mathbf{I}$.
2. Para $j = 0,\ldots,m-1 \;\rightarrow\; \mathbf{E}_{j+1} = -(\chi_{\{j\neq 0\}}\mathbf{E}_{j-1}v_{j-1}^+ - \mathbf{E}_j v_j^0)(v_{j+1}^-)^{-1}$.
3. Se resuelve $\mathbf{e}_0^T$ usando $\mathbf{e}_0^T(\mathbf{E}_m v_m^0 + \mathbf{E}_{m-1}v_{m-1}^+) = (0,\ldots,0,1)$.
4. Para $j = 1,\ldots,m \;\rightarrow\; \mathbf{e}_j^T = \mathbf{e}_0^T\mathbf{E}_j$
5. Normalización $\rightarrow\; \mathbf{e}_j^T = \frac{\mathbf{e}_j^T}{\sum_{j=0}^m \mathbf{e}_j^T\mathbf{1}}$

En principio el algoritmo 1 se presenta como una mejor solución para la resolución del sistema puesto que su coste computacional va a ser inferior, sin embargo este algoritmo puede presentar algún que otro problema, de hecho en el artículo [Ser02] se plantea la posibilidad de que la operación v_{j+1}^{-1} del Paso 2, no se pueda llevar a cabo por un problema de singularidad. Es por ello que se ha optado por el uso del algoritmo 0.

C.3 Value Extrapolation. Función de extrapolación

El objetivo de Value Extrapolation, VE, es encontrar una función de extrapolación, $f(s)$, que ajuste algunos puntos (s, w_s) en $s \in \hat{\mathcal{S}}$ de modo que aproxime también (s, w_s) en $s \notin \hat{\mathcal{S}}$. Una de las funciones más utilizadas en ese sentido son las funciones polinómicas. Obviamente, la función f_s, así como el conjunto de puntos donde se evalúa debe escogerse de modo no ambiguo, es decir, en el caso de tomar un ajuste polinómico, como es nuestro caso, el número de puntos tomados para evaluar la función debe ser igual o superior al número de coeficientes del polinomio.

De este modo, si K es el número de dimensiones del proceso de Markov y n_i es el grado de interpolación polinómica deseado para la dimensión i, el polinomio de extrapolación buscado tendrá la forma:

$$f_s = \sum_{i=1}^{K} \sum_{j=0}^{n_i} a_{i,j} s_i^j$$

Para calcular los parámetros $a_{i,j}$ el proceso de ajuste tratará de minimizar el error cuadrático

$$E = \sum_{s \in \mathcal{S}_f} (f_s - w_s)^2 = \sum_{s \in \mathcal{S}_f} (\sum_{i=1}^{K} \sum_{j=0}^{n_i} a_{i,j} s_i^j - w_s)^2 \qquad \text{(C.9)}$$

De forma que los parámetros óptimos se puedan calcular resolviendo las ecuaciones:

$$\frac{\partial E}{\partial a_{i,j}} = 0 \quad \forall i, j$$

Este procedimiento resulta poco práctico. Una alternativa sería utilizar la base de Lagrange. En este caso, dado un conjunto de puntos, tal que

$$n+1 = |\mathcal{S}_f| \quad \text{puntos} \quad (s_0, w_{s_0}), \ldots, (s_n, w_{s_n}),$$

donde no hay dos s_j idénticos, el polinomio de interpolación en la forma de Lagrange es una combinación lineal

$$f_s = L(s) := \sum_{j=0}^{n} v_{s_j} \ell_j(s)$$

de polinomios base de Lagrange

$$\ell_j(s) \;:=\; \prod_{i=0,\, i\neq j}^{n} \frac{s-s_i}{s_j-s_i} = \frac{s-s_0}{s_j-s_0} \cdots \frac{s-s_{j-1}}{s_j-s_{j-1}} \frac{s-s_{j+1}}{s_j-s_{j+1}} \cdots \frac{s-s_n}{s_j-s_n} \quad \text{(C.10)}$$

donde puede verse fácilmente que, $\ell_j(s)$ es un polinomio de grado n y $\ell_i(s_j) = \delta_{ij}, \quad 0 \leq i, j \leq n$, siendo δ_{ij} la delta de Kronecker.

Se necesita extrapolar el punto w_{Q+1}. Para ello partimos del polinomio de extrapolación que expresaremos como:

$$f_s = a_0 + a_1(s-s_0) + a_2(s-s_0)(s-s_1) + a_3(s-s_0)(s-s_1)(s-s_2) + \ldots$$

En nuestro caso:

$$f_s = a_0 + a_1(s-Q) + a_2(s-Q)(s-Q+1) + a_3(s-Q)(s-Q+1)(s-Q+2) + \ldots$$

de forma que:

$$\begin{aligned}
w_Q = f_Q &= a_0, \\
w_{Q-1} = f_{Q-1} &= a_0 - a_1, \\
w_{Q-2} = f_{Q-2} &= a_0 - 2a_1 + 2a_2, \\
w_{Q-3} = f_{Q-3} &= a_0 - 3a_1 + 3*2a_2 - 3*2*1a_3, \\
w_{Q-4} = f_{Q-4} &= a_0 - 4a_1 + 4*3a_2 - 4*3*2a_3 + 4*3*2*1a_4, \\
&\ldots
\end{aligned}$$

Luego:

$$\begin{aligned}
a_0 =& w_Q,\\
a_1 =& w_Q - w_{Q-1},\\
a_2 =& \frac{w_Q - 2w_{Q-1} + w_{Q-2}}{2!},\\
a_3 =& \frac{w_Q - 3w_{Q-1} + 3w_{Q-2} - w_{Q-3}}{3!},\\
a_4 =& \frac{w_Q - 4w_{Q-1} + 6w_{Q-2} - 4w_{Q-3} + w_{Q-4}}{4!},\\
&\ldots
\end{aligned}$$

Por lo tanto la interpolación de grado n del punto w_{Q+1} se puede expresar como:

$$\begin{aligned}
w_{Q+1}^{(n)} =& f(Q+1) = a_0 + a_1 + 2!a_2 + 3!a_3 + 4!a_4 + \ldots =\\
=& w_Q + (w_Q - w_{Q-1}) + (w_Q - 2w_{Q-1} + w_{Q-2}) + (w_Q - 3w_{Q-1} +\\
&+3w_{Q-2} - w_{Q-3}) + (w_Q - 4w_{Q-1} + 6w_{Q-2} - 4w_{Q-3} + w_{Q-4}) + \ldots =\\
=& \binom{n+1}{1} w_Q - \binom{n+1}{2} w_{Q-1} + \binom{n+1}{3} w_{Q-3} - \binom{n+1}{4} w_{Q-4} + \ldots
\end{aligned}$$

De donde se llega a

$$w_{Q+1}^{(n)} = \sum_{k=0}^{n} (-1)^k \binom{n+1}{k+1} w_{Q-k} \tag{C.11}$$

C.3.1 Ejemplos

Por ejemplo, en el caso de extrapolación lineal el polinomio de ajuste tiene la forma $f(x) = ax + b$. En este caso, los puntos usados son $(Q, w_{(C,Q)})$ y $(Q-1, w_{(C,Q-1)})$. Por lo tanto, se tiene el siguiente polinomio de interpolación:

$$\begin{aligned}
w_{(C,s)} &= w_{(C,Q)} \frac{(s - Q + 1)}{1} + w_{(C,Q-1)} \frac{s - Q}{-1} =\\
&= (w_{(C,Q)} - w_{(C,Q-1)})s + w_{(C,Q)}(1 - Q) + Q w_{(C,Q-1)}
\end{aligned}$$

Reemplazando $s = Q + 1$ para calcular $w_{(C,Q+1)}$, se tiene:

$$w^{(1)}_{(C,Q+1)} = 2w_{(C,Q)} - w_{(C,Q-1)}.$$

Siguiendo un procedimiento similar al seguido en la extrapolación lineal, para un polinomio cuadrático se obtiene la siguiente relación:

$$w^{(2)}_{(C,Q+1)} = 3w_{(C,Q)} - 3w_{(C,Q-1)} + w_{(C,Q-2)}$$

Y para el ajuste a un polinomio cúbico:

$$w^{(3)}_{(C,Q+1)} = 4w_{(C,Q)} - 6w_{(C,Q-1)} + 4w_{(C,Q-2)} - w_{(C,Q-3)}.$$

C.4 Cálculo de los parámetros para el modelo de dos órbitas

Balanceando el flujo de probabilidades que entran y salen de un determinado conjunto de estados se pueden calcular los diferentes parámetros de la aproximación. Para calcular M_n y p_n es necesario definir las transiciones entre aquellos estados que mantienen m constante, así tendremos una serie de Q_n ecuaciones, una para cada valor de m, de forma que:

$$\sum_{x \in S_a, y \in S_b} q_{xy}\pi_x = \sum_{x \in S_a, y \in S_b} q_{yx}\pi_y$$

donde los subespacios S_a y S_b, se definen como

$$S^{(i)}_a = \{(k,m,s) : 0 \leq k \leq C; m = i-1; 0 \leq s \leq Q_h$$
$$S^{(i)}_b = \{(k,m,s) : 0 \leq k \leq C; m = i; 0 \leq s \leq Q_h$$

para $i \in [1, Q_n]$.

Se obtendrá una ecuación de balance diferente para cada valor de i. Así tendremos:

- $i = 1$

$$\mu_{red} \sum_{k=0}^{L-1} \sum_{s=0}^{Q_h} \pi(k,2,s) + f\mu_{red} \sum_{s=0}^{Q_h} \pi(L,2,s) +$$
$$+\mu_{red}(1-f)P_{in} \sum_{s=0}^{Q_h} \pi(L,2,s) + +\mu_{red}P_{in} \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,2,s) =$$
$$= \lambda_n(1-f)(1-P_{in}^1) \sum_{s=0}^{Q_h} \pi(L,0,s) + \lambda_n(1-P_{in}^1) \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,0,s)$$

- $i = 2$

$$2\mu_{red} \sum_{k=0}^{L-1} \sum_{s=0}^{Q_h} \pi(k,1,s) + 2f\mu_{red} \sum_{s=0}^{Q_h} \pi(L,1,s) +$$
$$+2\mu_{red}(1-f)P_{in} \sum_{s=0}^{Q_h} \pi(L,1,s) + +2\mu_{red}P_{in} \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,1,s) =$$
$$= \lambda_n(1-f)(1-P_{in}^1) \sum_{s=0}^{Q_h} \pi(L,1,s) + \lambda_n(1-P_{in}^1) \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,1,s)$$

- ...

- $i = Q_n$

$$\alpha_n \sum_{k=0}^{L-1} \sum_{s=0}^{Q_h} \pi(k,Q_n,s) + f\alpha_n \sum_{s=0}^{Q_h} \pi(L,Q_n,s) +$$
$$+\alpha_n(1-f)P_{in} \sum_{s=0}^{Q_h} \pi(L,Q_n,s) + \alpha_n P_{in} \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,Q_n,s) =$$
$$= \lambda_n(1-f)(1-P_{in}^1) \sum_{s=0}^{Q_h} \pi(L,Q_n-1,s) +$$
$$+\lambda_n(1-P_{in}^1) \sum_{k=L+1}^{C} \sum_{s=0}^{Q_h} \pi(k,Q_n-1,s)$$

Sumando todas las ecuaciones de balance se obtiene:

$$\mu_{red}\sum_{k=0}^{L-1}\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h} m\pi(k,m,s)+f\mu_{red}\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h} m\pi(L,m,s)+\mu_{red}(1-f)P_{in}\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h} m\pi(L,m,s)+$$
$$+\mu_{red}P_{in}\sum_{k=L+1}^{C}\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h} m\pi(k,m,s)+\alpha_n\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+f\alpha_n\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+$$
$$+\alpha_n(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+\alpha_n P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)=$$
$$=\lambda_n(1-f)(1-P_{in}^1)\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h}\pi(L,m,s)+\lambda_n(1-P_{in}^1)\sum_{k=L+1}^{C}\sum_{m=0}^{Q_n-1}\sum_{s=0}^{Q_h}\pi(k,m,s)$$

Reordenando esta última ecuación y teniendo en cuenta que la tasa de peticiones nuevas que reintentan es igual a la suma de las tasas de reintentos exitosos más la de reintentos que abandonan el sistema sin obtener servicio, se obtiene:

$$\lambda_n(1-f)(1-P_{in}^1)\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+\lambda_n(1-P_{in}^1)\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)=$$
$$=p_n M_n\mu_{red}\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+p_n f M_n\mu_{red}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+$$
$$+p_n M_n\mu_{red}(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+p_n M_n\mu_{red}P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s) \qquad \text{(C.12)}$$

Por otro lado, de la última ecuación de balance ($i = Q_n$) se tiene:

$$\lambda_n(1-f)(1-P_{in}^1)\sum_{s=0}^{Q_h}\pi(L,Q_n-1,s)+\lambda_n(1-P_{in}^1)\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n-1,s)=$$
$$=M_n\mu_{red}\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+fM_n\mu_{red}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+M_n\mu_{red}(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+$$
$$+M_n\mu_{red}P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)-p_n M_n\mu_{red}\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)-$$
$$-p_n f M_n\mu_{red}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)-p_n M_n\mu_{red}(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)-$$
$$-p_n M_n\mu_{red}P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s) \qquad \text{(C.13)}$$

De forma que sumando (C.12) y (C.13) obtenemos:

$$\lambda_n(1-P_{in}^1)\Big[\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}[\pi(k,Q_n-1,s)+\pi(k,Q_n,s)]+$$
$$+(1-f)\sum_{s=0}^{Q_h}[\pi(L,Q_n-1,s)+\pi(L,Q_n,s)]\Big]=$$
$$M_n\mu_{red}\Big[\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+f\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+$$
$$+P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)\Big]$$

A partir de esta expresión se puede despejar M_n obteniendo

$$M_n=\frac{\lambda_n(1-P_{in}^1)\zeta_2}{\mu_{red}\zeta_3}$$

con:

$$\zeta_2=\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}[\pi(k,Q_n-1,s)+\pi(k,Q_n,s)]+(1-f)\sum_{s=0}^{Q_h}[\pi(L,Q_n-1,s)+\pi(L,Q_n,s)]$$
$$\zeta_3=\sum_{k=0}^{L-1}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+f\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+(1-f)P_{in}\sum_{s=0}^{Q_h}\pi(L,Q_n,s)+P_{in}\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)$$

Y volviendo a la ecuación (C.12) y sustituyendo M_n por la expresión calculada obtenemos p_n como $p_n=\frac{\zeta_1}{\zeta_2}$ donde ζ_2 es la misma que la utilizada en M_n, mientras que ζ_1 se corresponde con:

$$\zeta_1=\sum_{k=L+1}^{C}\sum_{s=0}^{Q_h}\pi(k,Q_n,s)+(1-f)\sum_{s=0}^{Q_h}\pi(L,Q_n,s)$$

Los parámetros M_h y p_h se calculan de forma análoga. En este caso los subespacios a considerar son aquellos en que el valor de s es constante, apareciendo, por tanto, Q_h ecuaciones de balance:

- $j=1$

$$\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,1)+\mu_{ret}P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,1)=\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,0)$$

- $j = 2$

$$2\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,2)+2\mu_{ret}P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,2)=\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,1)$$

- ...

- $j = Q_h$

$$\alpha_h\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+\alpha_h P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)=\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,Q_h-1)$$

Siguiendo un procedimiento análogo al utilizado para calcular p_n y M_n, sumamos las ecuaciones de balance:

$$\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h-1}\pi(C,m,s)=\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h-1}s\pi(k,m,s)+\mu_{ret}P_{ih}\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h-1}s\pi(C,m,s)+$$
$$+M_h\mu_{ret}(1-p_h)\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+M_h\mu_{ret}(1-p_h)P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)$$

y reordenando la expresión, tenemos:

$$\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h}\pi(C,m,s)-\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,Q_h)=\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h-1}s\pi(k,m,s)+$$
$$+M_h\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)-p_hM_h\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+\mu_{ret}P_{ih}\sum_{m=0}^{Q_n}\sum_{s=0}^{Q_h-1}s\pi(C,m,s)+$$
$$+M_h\mu_{ret}P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Qh)-p_hM_hP_{ih}\mu_{ret}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)$$

Teniendo en cuenta que la tasa de primeras peticiones de *handovers* bloqueados es igual a la suma de la tasa de reintentos automáticos de *handovers* que resultan exitosos más aquellos que abandonan el sistema sin ser servidos, se llega a:

$$\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,Q_h)=p_hM_h\mu_{ret}\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+p_hM_hP_{ih}\mu_{ret}\sum_{m=0}^{Q_n}\pi(C,m,Q_h) \quad \text{(C.14)}$$

Considerando la última ecuación de balance ($j = Q_h$):

$$\lambda_h(1-P_{ih}^1)\sum_{m=0}^{Q_n}\pi(C,m,Q_h-1)=\mu_{ret}\Big[M_h\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)-p_hM_h\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+ \\ +M_hP_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)-p_hM_hP_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)\Big] \tag{C.15}$$

Sumando las ecuaciones (C.14) y (C.15):

$$\lambda_h(1-P_{ih}^1)\Big[\sum_{m=0}^{Q_n}\pi(C,m,Q_h)+\sum_{m=0}^{Q_n}C,m,Q_h-1\Big]= \\ =M_h\mu_{ret}\Big[\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)\Big]$$

De donde se puede obtener directamente M_h. Para obtener p_h solo será necesario llevar la expresión de M_n a la ecuación (C.14), obteniéndose respectivamente:

$$p_h=\frac{\sum_{m=0}^{Q_n}\pi(C,m,Q_h)}{\sum_{m=0}^{Q_n}\big[\pi(C,m,Q_h)+\pi(C,m,Q_h-1)\big]} \tag{C.16}$$

$$M_h=\frac{\lambda_h(1-P_{ih}^1)\Big[\sum_{m=0}^{Q_n}\big[\pi(C,m,Q_h)+\pi(C,m,Q_h-1)\big]}{\mu_{ret}\Big[\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n}\pi(k,m,Q_h)+P_{ih}\sum_{m=0}^{Q_n}\pi(C,m,Q_h)\Big]} \tag{C.17}$$

C.5 Esquema de control de admisión dinámico

La figura C.2 muestra el funcionamiento de la política de control de admisión dinámico utilizado. Como se puede observar para admitir una petición se comprueba si existen, por lo menos c_i unidades de recurso libres. Nótese que una vez verificado que se cumple esta condición, será necesario comprobar

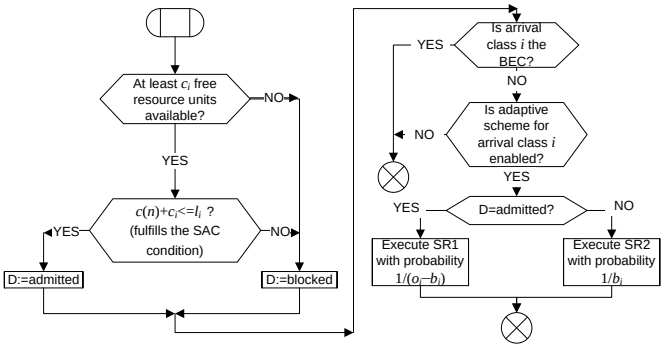
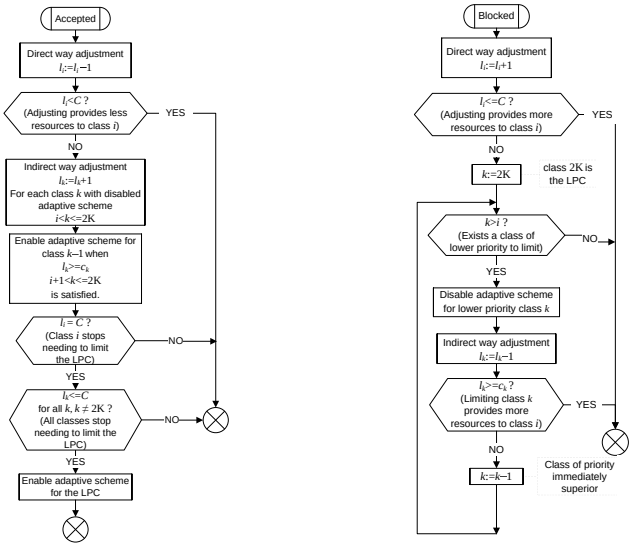


Figura C.2: Mecanismo de control de admisión dinámico

que se cumple la condición de aceptación impuesta por el mecanismo de control de admisión.

Una vez tomada la decisión de admisión, ya sea de aceptación o de rechazo, el esquema dinámico reajusta el umbral correspondiente. Para realizar este ajuste se utiliza un mecanismo probabilístico basado en el objetivo de bloqueo establecido para dicho flujo. Este mecanismo se puede describir mediante las subrutinas SR1 y SR2 que se muestran en la figura C.3.

Expuesto de una forma sencilla, el esquema dinámico se puede entender como compuesto por un conjunto de procesos, uno por cada uno de los flujos existentes, que trabajan en paralelo. En condiciones normales cada unos de esos procesos trabaja de forma independiente a los demás, sin embargo cuando alguno de los flujos sufre congestión. En este caso los procesos de los flujos de menor prioridad pasan a estar controlados por el proceso del flujo que experimenta la congestión. Nótese que el umbral del flujo BEC no se actualiza cuando el control de admisión toma decisiones respecto a peti-



(a) Rutina SR1: ajuste de umbrales tras una aceptación.

(b) Rutina SR2: ajuste de umbrales tras un rechazo.

Figura C.3: Algoritmo de ajuste del control de admisión dinámico

ciones de este flujo. La rutina SR2 gestiona las acciones a realizar cuando el umbral correspondiente sufre un incremento, es decir tras una decisión de bloqueo. Por lo tanto será el encargado de activar el modo indirecto de funcionamiento del esquema y así, de deshabilitar los procesos de los flujos de menor prioridad. Por contra la rutina SR1 es la encargada de gestionar las acciones a realizar cuando se disminuye el umbral, es decir, tras una decisión de aceptación.

Apéndice D

Publicaciones

D.1 Relacionadas con la tesis

D.1.1 Capítulo de libro

1. Jorge Martínez-Bauset, Vicent Pla, David García, Mª José Doménech, José Manuel Giménez-Guzmán.
   Título del libro: **Advances in Wireless Networks: Performance Modelling, Analysis and Enhancement**.
   Título del capítulo: **Designing admission control policies to minimize blocking/forced-termination**.
   Volume 8 in Wireless Networks and Mobile Computing (Yi Pan - Editor). ISBN: 1-60021-713-3. Nova Science Publishers.

D.1.2 Revista

1. Mª José Doménech-Benlloch, José Manuel Giménez-Guzmán, Jorge Martínez-Bauset, Vicente Casares-Giner.
   Efficient and accurate methodology for solving multiserver retrial

systems, IEE Electronic Letters. Vol. 41, no. 17, pp. 967-969. Agosto 2005. ISSN 0013-5194.

2. David Garcia-Roger,Mª Jose Domenech-Benlloch, Jorge Martinez-Bauset, and Vicent Pla.
   Adaptive trunk reservation policies in multiservice mobile wireless networks. In Proceedings of the 8th International Conference on Management of Multimedia Networks and Services (MMNS'05), Lecture Note in Computer Science (LNCS), Springer. Vol. 3754, pp. 47 — 58, Octubre 2005.

3. Mª José Doménech-Benlloch, José Manuel Giménez-Guzmán, Jorge Martínez Bauset, Vicente Casares-Giner.
   A Low Computation Cost Algorithm to Solve Cellular Systems with Retrials Accurately, Wireless Systems and Network Architectures in Next Generation Internet, Lecture Notes in Computer Science (LCNS), Springer. Vol. 3883, pp. 103-114. Febrero 2006.

4. David Garcia-Roger,Mª Jose Domenech-Benlloch, Jorge Martinez-Bauset, and Vicent Pla.
   Hierarchical admission vontrol in mobile cellular networks using adaptive bandwidth reservation,Wireless Systems and Network Architectures in Next Generation Internet, Lecture Notes in Computer Science (LCNS), Springer. Vol. 3883, pp. 130–144, Febrero 2006.

5. David Garcia-Roger,Mª Jose Domenech-Benlloch, Jorge Martinez-Bauset, and Vicent Pla.
   Esquema adaptativo de reserva para el control de admisión jerárquico en redes móviles celulares, IEEE Latin America Transactions, Vol. 4, no. 6, pp. 392 – 398, Diciembre 2006.

6. Jose Manuel Gimenez-Guzman,Mª Jose Domenech-Benlloch, Vicent Pla, Vicente Casares-Giner, Jorge Martinez-Bauset.
   Analysis of a Cellular Network with User Redials and Automatic Network Retrials, The 7th International Conference on Next Generation Teletraffic and Wired / Wireless Advanced Networking (NEW2AN 2007), Lecture Notes in Computer Science (LCNS), Springer, vol. 4712, pp. 210–222, Septiembre 2007.

7. Jose Manuel Gimenez-Guzman, Mª Jose Domenech-Benlloch, Vicent Pla, Vicente Casares-Giner, Jorge Martinez-Bauset.
   Efecto de los remarcados y reintentos automáticos en redes celulares, IEEE Latin America Transactions, Vol. 5, no. 6, pp. 433-440. Octubre 2007. ISSN 1548-0992.
   Escogido a partir de la versión presentada en:
   VI Jornadas de Ingeniería Telemática (JITEL 2007). Málaga. Septiembre 2007.

8. D. Garcia-Roger, Mª Jose Domenech-Benlloch, J. Martinez-Bauset, V. Pla.
   Adaptive Admission Control in Mobile Cellular Networks with Streaming and Elastic Traffic, The 20th International Teletraffic Congress (ITC), Ottawa, Canada, Sunday 17th - Thursday 21th, July 2007. Lecture Notes in Computer Science (LNCS), Springer-Verlag, Vol. 4516, pp. 925-937, Septiembre 2007.

9. Mª José Doménech-Benlloch, José Manuel Giménez-Guzmán, Vicent Pla, Jorge Martínez-Bauset, Vicente Casares-Giner.
   Generalized Truncated Methods for an Efficient Solution of Retrial Systems, Mathematical Problems in Engineering, Vol. 2008, Article ID 183089, pp. 1-15. 2008.

10. José Manuel Giménez-Guzmán, Mª José Doménech-Benlloch, Vicent Pla, Vicente Casares-Giner, Jorge Martínez-Bauset.
    Guaranteeing Seamless Mobility with User Redials and Automatic Handover Retrials, Journal of Universal Computer Science. Vol. 14, no. 10, pp. 1597-1624. May 2008.

11. José Manuel Giménez-Guzmán, Mª José Doménech-Benlloch, Vicent Pla, Vicente Casares-Giner, Jorge Martínez-Bauset.
    Value Extrapolation Technique to Solve Retrial Queues: a Comparative Perspective, ETRI Journal, Vol.30, no.3, pp.492-494. June 2008.

12. Jorge Martínez-Bauset, David García-Roger, Mª José Doménech-Benlloch, Vicent Pla.
    Maximizing the capacity of mobile cellular networks with heterogeneous traffic, Computer Networks, Vol.59,no.7, pp. 973-988, 2009.

D.1.3 Congreso

Internacional

1. David Garcia-Roger, Mª Jose Domenech-Benlloch, Jorge Martinez-Bauset, and Vicent Pla.
   Adaptive admission control scheme for multiservice mobile cellular networks. EuroNGI First Conference on Next Generation Internet: Traffic Engineering (NGI), pp. 288–295, Apr. 2005.

2. David Garcia-Roger, Mª Jose Domenech-Benlloch, JorgeMartinez-Bauset, and Vicent Pla.
   Comparative evaluation of adaptive trunk reservation schemes for mobile vellular networks. In Proceedings HET-NETs ’05 - Performance

Modelling and Evaluation of Heterogeneous Networks. Ilkley, United Kingdom. Julio 2005.

3. José Manuel Giménez-Guzmán, Mª José Doménech-Benlloch, Jorge Martínez-Bauset, Vicent Pla, Vicente Casares-Giner.
   Analysis of a Handover Procedure with Queueing, Retrials and Impatient Customers, In Proceedings HET-NETs '05 - Performance Modelling and Evaluation of Heterogeneous Networks. Ilkley, United Kingdom. Julio 2005.

4. José Manuel Giménez-Guzmán, Mª José Doménech-Benlloch, Vicent Pla, Vicente Casares-Giner, Jorge Martínez-Bauset.
   Efficient and Accurate Solution of Multiserver Retrial Systems with User Impatience Through the Value Extrapolation Technique, 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, June 16-18, 2008, Edinburgh, UK.

Nacional

1. Mª José Doménech-Benlloch, José Manuel Giménez-Guzmán, Vicente Casares-Giner.
   Modelos Markovianos para la resolución de sistemas con reintentos. Evaluación de diferentes metodologías, IV Jornadas de Ingeniería Telemática (Jitel'03). Gran Canaria, España. Septiembre 2003.

2. David García Roger,Mª José Doménech Benlloch, JorgeMartínez Bauset, y Vicent Pla.
   Esquema adaptativo de reserva para redes móviles celulares, V Jornadas de Ingeniería Telemática (Jitel'05), pp. 25–32, Vigo, España, Septiembre 2005.

3. David García Roger, Mª José Doménech Benlloch, JorgeMartínez Bauset, y Vicent Pla.
   Esquema adaptativo para el control de admisión en redes celulares multiservicio, XV Jornadas Telecom I+D, noviembre 2005.

D.2 Otras publicaciones

D.2.1 Congreso

Internacional

1. R. Cosma, A. Cabellos-Aparicio, Mª J. Domenech-Benlloch, J.M. Gimenez-Guzman, J. Martinez-Bauset, M. Cristian, A. Fuentetaja, A. López, J. Domingo-Pascual, J. Quemada.
   Measurement-Based Analysis of the Performance of several Wireless Technologies, The 16th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2008), Cluj-Napoca, Transylvania, Romania. Septiembre 2008.

Apéndice E

Ámbito de la Tesis

Este trabajo se ha realizado en el ámbito de los siguientes proyectos de investigación de caracter nacional:

- Soporte a la calidad de servicio (QoS) extremo a extremo en redes multiservicio basadas en IP que permiten movilidad y servicios multimedia (Ministerio de Ciencia y Tecnología. TIC2001-0956-C04-04)
- Control de Admisión en Redes Móviles Heterogéneas (Ministerio de Educación y Ciencia. TSI2005-07520-C03-03)
- Admission Control in Mobile Networks with Rate-Adaptive Streams and Hierarchical Architecture (Ministerio de Ciencia e Innovación. TIN2008-06739-C04-02 / TSI)

Asimismo se agradece el apoyo a este trabajo por parte la Unión Europea, a través de la red de excelencia europea Design and Engineering of the Next Generation Internet (EuroNGI), y las contiuaciones de la misma, euroFGI y euroNF.

Por último agradecer el apoyo a este trabajo por parte del Gobierno Español a través de la beca de formación de profesorado univeristario, FPU, con referencia AP-2004-3332.

Bibliografía

[AA02] V. V. Anisimov y J. R. Artalejo, *Approximation of multiserver retrial queues by means of generalized truncated models*, TOP (Journal of Operations Research) **10** (2002), no. 1, 51–66.

[AG08] N. Argiriou y L. Georgiadis, *A framework for providing user level quality of service guarantees in multi-class rate adaptive systems*, Journal of Network Systems Management **16** (2008).

[AGC08] J. R. Artalejo y A. Gómez-Corral, *Retrial queueing systems, a computational approach*, Springer, 2008.

[AL02] A. S. Alfa y W. Li, *PCS networks with correlated arrival process and retrial phenomenon*, IEEE Transactions on Wireless Communications **1** (2002), no. 4, 630–637.

[AP02] J. R. Artalejo y M. Pozo, *Numerical calculation of the stationary distribution of the main multiserver retrial queue*, Annals of Operations Research **116** (2002), no. 1–4, 41–56.

[Art96] J. R. Artalejo, *Stationary analysis of the characteristics of the M/M/2 queue with constant repeated attempts*, Opsearch **33** (1996), 83–95.

[Art99a] _____, *Accessible bibliography on retrial queues*, Mathematical and Computer Modelling **30** (1999), 1–6.

[Art99b] _____, *A classified bibliography of research on retrial queues: Progress in 1990–1999*, TOP **7** (1999), 187–211.

[Bar04] F. Barceló, *Performance analysis of handoff resource allocation strategies through the state-dependent rejection scheme*, IEEE Transactions on Wireless Communications **3** (2004), no. 3, 900–909.

[BDK89] P. Boyer, A. Dupuis, y A. Khelladi, *A simple model for repeated calls due to time-outs*, Proceedings of the 12th International Teletraffic Congress, 1989, pp. 356–363.

[BR03] T. Bonald y J. Roberts, *Congestion at flow level and the impact of user behaviour*, Computer Networks **42** (2003), 521–536.

[BS97] S. K. Biswas y B. Sengupta, *Call admissibility for multirate traffic in wireless ATM networks*, Proc. 16th Ann. Joint Conf. IEEE Comp. & Comm. Soc. (INFOCOM), vol. 2, 1997, pp. 649–657.

[BT97] L. W. Bright y P. G. Taylor, *Equilibrium distributions for level-dependent quasi-birth-and-death processes*, Lecture notes in Pure and Applied Maths **183** (1997), 359–375.

[BVE99] C. Bettstetter, H.-J. Vögel, y J. Eberspächer, *GSM phase 2+ general packet radio service GPRS: architecture, protocols, and air interface*, IEEE Communication Surveys **2** (1999), no. 3, 2 – 14.

[BXG07] W. Bolton, Y. Xiao, y M. Guizani, *IEEE 802.20: mobile broad-band wireless access*, IEEE Wireless Communications **14** (2007), 84–95.

[Cas01] V. Casares, *Variable bit rate voice using hysteresis thresholds*, Telecommunication Systems **17** (2001), no. 1,2, 31–62.

[CC97] C. Chao y W. Chen, *Connection admission control for mobile multiple-class personal communications networks*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 8, 1618 – 1626.

[CCL95] B. D. Choi, K. B. Choi, y Y. W. Lee, *M/ G /1 retrial queueing systems with two types of calls and finite capacity*, Queueing Systems **19** (1995), no. 1-2, 215–229.

[Cho92] B. D. Choi, *Retrial queues with collision arising from unslotted CS-MA/CD protocol*, Queueing Systems **11** (1992), 335–356.

[CKJ06] S. R. Chakravarthy, A. Krishnamoorthy, y V. Joshua, *Analysis of a multi-server retrial queue with search of customers from the orbit*, Performance Evaluation **63** (2006), no. 8, 776–798.

[Coh57] J.W. Cohen, *Basic problems of telephone traffic theory and the influence of repeated calls*, Phillips Telecommunication Review **18** (1957), 49–100.

[CPOG04] F. A. Cruz-Perez y L. Ortigoza-Guerrero, *Flexible resource allocation strategies for class-based QoS provisioning in mobile networks*, IEEE Trans. on Vehicular Technology **53** (2004), no. 3, 805 – 819.

[CS02] C-T. Chou y K.G. Shin, *Analysis of combined adaptive bandwidth allocation and admission control in wireless networks*, Proceeding of IEEE INFOCOM (New York, USA), 2002, pp. 676 – 684.

[Ekl86] B. Eklundh, *Channel utilization and blocking probability in a cellular mobile telephone system with directed retry*, IEEE Transactions on Communications **34** (1986), no. 4, 329–337.

[Esp03] Real Academia Española, *Diccionario de la lengua española (edición electrónica)*, Espasa Calpe, 2003.

[Fal83] G. I. Falin, *Calculation of probability characteristics of a multiline system with repeat calls*, Moscow University Computational Mathematics and Cybernetics **1** (1983), 43–49.

[Fal90] _____, *A survey on retrial queues*, Queueing Systems **7** (1990), no. 2, 127–168.

[FNO99] A. Furuskär, J.s Näslund, y H. Olofsson, *Edge—-enhanced data rates for GSM and TDMA/136 evolution*, Ericsson Review, 1999.

[FR79] A. Fredericks y G. Reisner, *Approximations to stochastic service systems, with an application to a retrial model*, The Bell System Technical Journal **58** (1979), no. 3, 557–576.

[FT97] G. I. Falin y J. G. C. Templeton, *Retrial queues*, Chapman and Hall, 1997.

[GB06] M. Ghaderi y R. Boutaba, *Call admission control in mobile cellular networks: a comprehensive survey*, Wireless Communications and Mobile Computing **6** (2006), no. 1, 69 – 93.

[GC06] A. Gómez-Corral, *A bibliographical guide to the ananysis of the retrial queues through matrix analytic techniques*, Annals of Opperation Research **141** (2006), 163–191.

[GCR99] A. Gómez-Corral y M.F. Ramalhoto, *The stationary distribution of a markovian process arising in the theory of multiserver retrial queueing systems*, Mathematical and Computer Modelling **30** (1999), 159–170.

[Gir92] B. Girod, *Psychovisual aspects of image communications*, Signal Processing **28** (1992), no. 3, 121–135.

[GJL84] D.P. Gaver, P.A. Jacobs, y G. Latouche, *Finite birth-and-death models in randomly changing environments*, Advances in Applied Probability **16** (1984), 715–731.

[GMP05] D. Garcia, J. Martinez, y V. Pla, *Admission control policies in multiservice cellular networks: optimum configuration and sensitivity*, Lecture Notes in Computer Science **3427** (2005), 121 – 135.

[GR07] D. García-Roger, *Contribución al control de admisión en redes móviles celulares multiservicio*, Tesis doctoral, 2007.

[Gre01] D. Green, *Level-independent and level-dependent qbds and their processes of departure*, The 1st Seoul International Workshop on Queueing Theory (tutorial), 2001.

[Gué87] R. A. Guérin, *Channel occupancy time distribution in a cellular radio system*, IEEE Transactions on Vehicular Technology **35** (1987), 89–99.

[GW87] B. Greenberg y R. W. Wolff, *An upper bound on the performance of queues with returning customers*, Journal of Applied Probability **24** (1987), no. 2, 466–475.

[Haj82] B. Hajek, *Birth-and-Death processes on the integers with phases and general boundaries*, J, Applied Probability **19** (1982), 488–499.

[Han87] T. Hanschke, *Explicit formulas for the characteristics of the M/M/2/2 queue with repeated attemps*, Journal of Applied Probability **24** (1987), 486–494.

[Hil79] M. T. Hill, *Telecommunications switching principles*, 2nd edition Cambridge, MA: MIT Press, 1979.

[HJPS98] J. Hartung, A. Jacquin, J. Pawlyk, y K. Shipley, *A real-time scalable video codec for collaborative aplications over packet networks*, Proceeding of ACM Multimedia (Bristol, UK), Sept 12-16 1998, pp. 419–426.

[HL98] D.J. Houck y W.S. Lai, *Traffic modeling and analysis of hybrid fiber-coax systems*, Computer Networks and ISDN Systems **30** (1998), 821–834.

[HR86] D. Hong y S. S. Rappaport, *Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures*, IEEE Transactions on Vehicular Technology **35** (1986), no. 3, 77 – 92.

[Ive06] V. B. Iversen, *Teletraffic engineering and network planning*, http://www.com.dtu.dk/education/34340/, 2006.

[Jab96] B. Jabbari, *Teletraffic aspects of evolving and next-generation networks*, IEEE Pers. Comm. **3** (1996), no. 6, 4–9.

[Jan97] G.K. Janssens, *The quasi-random input queueing system with repeated attempts as a model for a collision-avoidance star local area network*, IEEE Transactions on Communications **45** (1997), 360–364.

[JS70] G. Jonin y J. Sedol, *Telephone systems with repeated calls*, Proceedings of the 6th International Teletraffic Congress ITC'6, 1970, pp. 435.1–435.5.

[Kau81] J. S. Kaufman, *Blocking in a shared resource environment*, Transactions on Communications **29** (1981), no. 10, 1474–1481.

[Kau92a] _____, *Blocking in a completely shared resource environment with state dependent resource and residency requirements*, Proceeding of IEEE INFOCOM (Florence, Italy), mayo 1992, pp. 2224–2232.

[Kau92b] _____, *Blocking with retrials in a completely shared resource environment*, Performance Evaluation **15** (1992).

[KCBN03] T. Kwon, Y. Choi, C. Bisdikian, y M. Naghshineh, *Qos provisioning in wireless/mobile multimedia networks using an adaptive framework*, Wireless Networks **9** (2003), no. 1, 51 – 59.

[KCD02] T. Kwon, Y. Choi, y S.K. Das, *Bandwidth adaptation algorithms for adaptive multimedia services in mobile cellular networks*, Wireless Personal Communications **22** (2002), no. 3, 337 – 357.

[Kle75] L. Kleinrock, *Queueing systems, volume i: Theory*, Wiley Interscience, 1975.

[Kuc73] A. Kuczura, *The interrupted Poisson process as an overflow process*, The Bell System Technical Journal **52** (1973), no. 3, 437–448.

[KZ97] F. Khan y D. Zeghlache, *Effect of cell residence time distribution on the performance of cellular mobile networks*, Proc. 47rd IEEE Veh. Technol. Conf. (VTC'97), vol. 2, 1997, pp. 949—953.

[Lee91] W. C. Y. Lee, *Smaller cells for greater performance*, IEEE Communications Magazine **29** (1991), no. 11, 19 – 23.

[LLC98] B. Li, C. Lin, y S. T. Chanson, *Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks*, Wireless Networks Journal **4** (1998), no. 4, 279 – 290.

[LPV06] J. Leino, A. Penttinen, y J. Virtamo, *Flow level performance analysis of wireless data networks: A case study*, Proceeding of IEEE ICC (Istambul, Turkey), June 11-15 2006, pp. 961–966.

[LR93] G. Latouche y V. Ramaswami, *A logarithmic reduction algorithm for quasi-birth-and-death processes*, Journal of Applied Probability **30** (1993), 650–674.

[LR99] _____, *Introduction to matrix analytic methods in stochastic modeling*, ASA-SIAM, 1999.

[LV06] J. Leino y J. Virtamo, *An approximative method for calculating performance measures of markov processes*, Proceeding of the 1st international conference on Performance evaluation methodolgies and tools (Pisa, Italy), 2006.

[MCL+99] M. A. Marsan, G. De Carolis, E. Leonardi, R. L. Cigno, y M. Meo, *How many cells should be considered to accurately predict the performance of cellular networks?*, Proceedings of European Wireless'99 (Munich, Germany), Oct.6-8 1999.

[MCL+01] M. A. Marsan, G. De Carolis, E. Leonardi, R. Lo Cigno, y M. Meo, *Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials*, IEEE Journal on Selected Areas in Communications **19** (2001), no. 2, 332–346.

[MK00] W. Mohr y W. Konhäuser, *Access network evolution beyond third generation mobile communications*, IEEE Communications Magazine **38** (2000).

[MLK02] I.D. Moscholios, M.D. Logothetis, y G.K. Kokkinakis, *Connection dependent threshold model: A generalization of the erlang multiple rate loss model*, Performance Evaluation **48** (2002), no. 1.

[MP92] M. Mouly y M.-B. Pautet, *The GSM system for mobile communications*, Published by the authors, 1992.

[MVJ97] S. McCanne, M. Veterli, y V. Jacobson, *Low-complexity video coding for receiver-driven layered multicast*, IEEE Journal on Selected Areas in Communications **17** (1997), no. 6, 983–1001.

[Nes79] M. Nesenbergs, *A hybrid of Erlang B and C formulas and its applications*, Transactions on Communications **27** (1979), no. 1, 59–88.

[Neu81] M. F. Neuts, *Matrix-geometric solutions in stochastic models: An algorithmic approach*, The Johns Hopkins University Press, 1981.

[NR90] M. F. Neuts y B. M. Rao, *Numerical investigation of a multiserver retrial model*, Queueing Systems **7** (1990), 169–190.

[NWL97] M. Naghshined y M. Willebeek-LeMair, *End-to-end QoS provisioning in multimedia wireless/mobile networks using and adaptive framework*, IEEE Communications Magazine **35** (1997), no. 11, 72–81.

[ODEa02] E. Onur, H. Deliç, C. Ersoy, y M. U. Çaglayan, *Measurement-based replanning of cell capacities in gsm networks*, Computer Networks **39** (2002), no. 6, 749–767.

[PCG02a] V. Pla y V. Casares-Giner, *Analytical-numerical study of the handoff area sojourn time*, Proceedings of IEEE GLOBECOM, noviembre 2002, pp. 886–890.

[PCG02b] _____, *Effect of the handoff area sojourn time distribution on the performance of cellular networks*, Proceedings of IEEE MWCN, 2002, pp. 401–405.

[Pea89] C.E.M. Pearce, *Extended continued fractions, recurrence relations and two-dimensional markov processes*, Advances in Applied Probability **21** (1989), 357–375.

[Ram95] V. Ramaswami, *Matrix analytic methods: A tutorial overview with some extensions and results*, Proceedings of First international conference on matrix analytic methods in stochastic models (Flint, Michigan, U.S.A), 1995.

[RGF02] S. Racz, B. P. Gero, y G. Fodor, *Flow level performance analysis of a multi-service system supporting elastic and adaptive services*, Performance Evaluation **49** (2002), no. 1.

[RGS98] M. Ruggieri, F. Graziosi, y F. Santucci, *Modeling of the handover dwell time in cellular mobile communications systems*, IEEE Transactions on Vehicular Technology **47** (1998), no. 2, 489–498.

[Rob81] J. W. Roberts, *A service system with heterogeneous user requirements*, Performance of Data Communications Systems and their Applications (1981).

[RSAK99] P. Ramanathan, K. M. Sivalingam, P. Agrawal, y S. Kishore, *Dynamic resource allocation schemes during handoff for mobile multimedia wireless networks*, IEEE Journal on Selected Areas in Communications **17** (1999), no. 7, 1270 – 1283.

[RSK05] J. Roszik, J. Sztrik, y C.-S. Kim, *Retrial queues in the performance modeling of cellular mobile networks using Mosel*, International Journal of Simulation: Systems, Science and Technology **6** (2005), no. 1–2, 38–47.

[RTN97] R. Ramjee, D. Towsley, y R. Nagarajan, *On optimal call admission control in cellular networks*, Wireless Networks Journal **3** (1997).

[RWO95] S.M. Redl, M.K. Weber, y M.W. Oliphant, *An introduction to gsm*, Artech House, 1995.

[Sch05] G. Schembra, *A resource management strategy for multimedia adaptive-rate traffic in a wireless network with TDMA access*, IEEE Trans. on Wireless Communications **4** (2005), no. 1, 65 – 78.

[Ser02] L. D. Servi, *Algorithmic solutions to two-dimensional Birth-Death processes with application to capacity planning*, Telecommunication Systems **21** (2002), no. 2–4, 205–212.

[SJBD98] S. Sen, J. Jawanda, K. Basu, y S. Das, *Quality-of-service degradation strategies in multimedia wireless network*, Proceeding of IEEE Vehicular Technology Conference (Ottawa, Ontario, Canada), May 18–21 1998, pp. 1884–1888.

[SK97] G. Stamatelos y V. Koukoulidis, *Reservation-based bandwidth allocation in a radio ATM network*, IEEE / ACM Transactions on Networking **5** (1997), no. 3, 420–428.

[Sta07] Standard IEEE 802.16, *http://www.ieee802.org/16/pubs /p80216e.html*, 2007.

[Ste99] S. N. Stepanov, *Markov models with retrials: the calculation of stationary performance measures based on the concept of truncation*, Mathematical and Computer Modelling **20** (1999), 207–228.

[TB95] P. G. Taylor y L. Bright, *Calculating the equilibrium distribution of level dependent quasi-birth-and-death processes*, Communications in Statistics-Stochastic Models **11** (1995), no. 3, 497–525.

[TGM97] Phuoc Tran-Gia y Michel Mandjes, *Modeling of customer retrial phenomenon*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 8, 1406–1414.

[Tij86] H. C. Tijms, *Stochastic modelling and analysis. a computational approach*, John Wiley, 1986.

[Tij03] _____, *Algorithmic analysis of queues*, Wiley, Chichester, 2003.

[Wil56] R. Wilkinson, *Theories for toll traffic engineering in the usa*, The Bell System Technical Journal **35** (1956), no. 2, 421–507.

[WZZZ03] X.-P. Wang, J.-L. Zheng, W. Zeng, y G.-D. Zhang, *A probability-based adaptive algorithm for call admission control in wireless network*, Proceedings of the International Conference on Computer Networks and Mobile Computing (ICCNMC) (Shanghai, China), 2003, pp. 197—204.

[XCW02] Y. Xiao, C. L. P. Chen, y B. Wang, *Bandwidth degradation QoS provisioning for adaptive multimedia in wireless/mobile networks*, Computer Communications **25** (2002), no. 13, 1153–1161.

[YK02] O. Yu y S. Khanvilkar, *Dynamic adaptive QoS provisioning over GPRS wireless mobile links*, Proceedings of the IEEE International Conference on Communications (ICC) (New York, USA), 28 Apr. - 2 May 2002, pp. 1100—1104.

[YL94] J. Ye y S. Li, *Folding algorithm: A computational method for finite QBD processes with level-dependent transitions*, IEEE Transactions on Communications **42** (1994), no. 2/3/4, 625–639.

[YL97] O. Yu y V. Leung, *Adaptive resource allocation for prioritized call admission over an ATM-based wireless PCN*, IEEE Journal on Selected Areas in Communications **15** (1997), no. 7, 1208 – 1224.

[YT87] T. Yang y J. G. Templeton, *A survey on retrial queues*, Queueing Systems **2** (1987), no. 3, 201–233.

[ZL01] Y. Zhang y D. Liu, *An adaptive algorithm for call admission control in wireless networks*, Proceedings of the IEEE Global Communications Conference (GLOBECOM) (San Antonio, Texas, USA), 2001, pp. 3628—3632.