

Diseño óptimo de políticas de Control de Admisión (CA) del tipo *Multiple Fractional Guard Channel* (MFGC) para redes móviles multiservicio

Autor: Elena Bernal Mor

Director: Jorge Martínez Bauset

Resumen — La necesidad de garantizar Calidad de Servicio (QoS) en escenarios móviles multiservicio, lleva a explorar nuevas estrategias que permitan minimizar la probabilidad de terminación forzosa de una conexión debido a un fallo de handover por insuficiencia de recursos en la celda destino. Estas estrategias deben estar coordinadas con los sistemas de Control de Admisión (CA). Las políticas de CA del tipo MFGC son probablemente las que han suscitado un mayor interés pues consiguen unas buenas prestaciones y su implementación es bastante simple. Sin embargo, realizar el ajuste de parámetros de este tipo de políticas de una manera óptima entraña un esfuerzo computacional que puede llegar a ser prohibitivo. Para hacer frente a esta limitaciones computacionales se han propuesto algunos algoritmos basados en aproximaciones numéricas. De los resultados obtenidos, se observa que el coste computacional necesario puede llegar a ser muy elevado en algoritmos previos descritos en la literatura. La aproximación basada en fórmulas recurrentes no ofrece resultados totalmente convincentes, pues la configuración calculada a partir de éste no cumple los objetivos de QoS. Finalmente se propone un algoritmo que mejora los tiempos de cálculo respecto al algoritmo previo y obtiene resultados fiables, pero es mucho más lento que el algoritmo que se basa en la aproximación.

Abstract — The need to guarantee certain QoS requirements in multiservice cellular networks has led to explore new strategies to minimize blocking probabilities. These strategies must be coordinated with admission control systems (CA). In this paper we study Multiple Fractional Guard Channel (MFGC) policy which provides good results and its implementation is quite simple. Nevertheless, computing the optimal parameters setting of this policy can constitute a high computational cost. To face these computational limitations some algorithms based in numerical approximations are proposed. The obtained results show that the algorithm based in one-dimensional recursion formulas is not a good approximation since the policy configuration obtained do not fulfills the QoS constrains. But this approximation can be useful to accelerate a previous algorithm appeared in the literature. Finally, we propose another algorithm which outperforms this previous algorithm in terms of computational cost and fulfills the QoS constraints, but it is slower than the algorithm based in one-dimensional recursion formulas.

Autor: Elena Bernal Mor, email: elbermo@posgrado.upv.es

Director: Jorge Martínez Bauset, email: jmartinez@upvnet.upv.es

Fecha de entrega: 03-12-07

ÍNDICE

I. Introducción	3
II. Objetivo	5
III. Control de Admisión en redes celulares	6
III.1. Descripción del modelo.....	8
III.2. Políticas de Control de Admisión estáticas	10
III.3. Multiple Fractional Guard Channel (MFGC).....	12
IV. Implementación de MFGC	13
IV.1. Búsqueda de parámetros que ofrezcan capacidad óptima	13
IV.2. Aproximación mediante formulas recurrentes	18
IV.3. Algoritmo final propuesto.....	22
V. Comparativa de resultados y coste computacional	24
V.1. Escenario con dos tipos de servicio.....	24
V.2. Escenario con cuatro tipos de servicio	30
VI. Conclusiones	36
Agradecimientos	37
Referencias	37

I. INTRODUCCION

En los últimos años, las comunicaciones móviles han experimentado una revolución tanto tecnológica como de contenidos. Es sencillo ver que el negocio de la telefonía móvil se ha ido desarrollando alrededor de dos servicios básicos, la voz y los mensajes cortos. Pero la evolución de estos sistemas apunta hacia servicios con mucho más volumen de datos. El desarrollo y combinación de la informática, las tecnologías inalámbricas, la microelectrónica e Internet han llevado a la aparición de los servicios de datos de movilidad en banda ancha.

Las redes móviles se encuentran con un problema si pretenden hacer frente a la demanda que pueda originar este nuevo tipo de servicios junto con el crecimiento de usuarios móviles, ya que el ancho de banda del que disponen es limitado. Además no olvidemos que la voz y la mensajería no desaparecen y por tanto es necesario asegurar una convivencia entre éstos y los nuevos servicios de banda ancha.

Dado todo esto, resulta necesario garantizar una determinada calidad de servicio (QoS) a los clientes. Los parámetros de mérito que definen la QoS son de dos tipos: de nivel de paquete (retardo máximo extremo a extremo, variabilidad del retardo o *jitter* y pérdidas) y de nivel de sesión (probabilidades de bloqueo de sesiones nuevas y de *handover* y probabilidad de terminación forzosa). En nuestro caso trabajaremos a nivel de establecimiento de sesiones, por lo que se tendrán en cuenta este segundo tipo de parámetros para definir la QoS.

Como consecuencia de lo anteriormente comentado es preciso hacer una buena planificación de la red celular e implementar algún tipo de mecanismo que aproveche al máximo los recursos radio de los que dispone la red, teniendo en cuenta la convivencia de diferentes tecnologías, y de diferentes tipos de servicio que demandarán diferentes características de tráfico y de QoS.

A todo esto, se suman las características propias de las redes móviles. Es decir, todos los problemas que se pueden presentar debido a la propagación a través del canal radio y sobre todo el traspaso de llamadas de una celda a otra o *handovers*. A la hora de gestionar los flujos de llegada es necesario tener en cuenta que en una celda pueden aparecer tanto flujos de llegada nuevos, iniciados en esa misma celda, como flujos de llegada provenientes de otras celdas vecinas, que debido a la movilidad que permite el sistema alcanzan la celda en cuestión.

Para aumentar la capacidad del sistema se podría pensar como solución la reducción del tamaño de las celdas del sistema celular para reutilizar mejor el ancho de banda disponible, pero esto supondría aumentar el número de *handovers* entre celdas, lo cual tendría un gran impacto en cuanto a la gestión de los recursos radio y la QoS.

Para el usuario resulta más molesto el bloqueo de una llamada en curso debido al rechazo de una petición de *handover* que el rechazo de una sesión nueva. Por ese motivo se priorizan los *handovers*. Como se puede entrever, esta priorización y el modelo del sistema en general puede

llegar a ser bastante complejo, puesto que se tienen diferentes flujos de sesiones nuevos por un lado y de *handovers* por otro, teniendo cada uno de ellos unas necesidades de tráfico y por tanto unas restricciones en cuanto a QoS.

Gestionar los recursos radio de forma adecuada resulta un tema crucial, y el control de admisión de las sesiones resulta un mecanismo importante para esta gestión. Por todo ello, desde un tiempo atrás se han venido invirtiendo esfuerzos y trabajo en la búsqueda de un mecanismo de control de admisión de sesiones en entornos que presenten movilidad y múltiples servicios, de forma que por un lado se maximice el volumen de aceptación de flujos y por el otro lado se garanticen los niveles de QoS impuestos minimizando la probabilidad de bloqueo y de terminación forzosa en el caso de traspaso de llamada, que experimenta cada servicio debido a la saturación de los recursos radio.

Esta tarea puede resultar compleja no sólo por la búsqueda de la mejor política de admisión en cuanto a optimización de la capacidad de la red, sino también por el coste computacional que puede alcanzar la configuración de dicha política. En este sentido es necesario buscar un compromiso entre estos dos objetivos.

II. OBJETIVOS

En esta tesina se pretende analizar una de las políticas de Control de Admisión que ha dado mejores resultados, conocida como *Multiple Fractional Guard Channel* (MFGC). El estudio de esta política nos lleva al ajuste de ciertos parámetros que maximicen la capacidad del sistema pero al mismo tiempo, como se ha señalado anteriormente, es necesario minimizar el coste computacional que ello puede entrañar.

Para ello se analiza un primer algoritmo desarrollado en [1] que calcula la configuración óptima. Se analizan los resultados obtenidos al utilizar una aproximación por formulas recurrentes tal y como se señala en [2], y por último, se estudia la posibilidad de utilizar esta aproximación dentro del algoritmo de [1] para reducir tiempos de cálculo.

Todo ello se plasmará en dos escenarios diferentes que se detallan más adelante.

Finalmente se comparan los resultados obtenidos de las distintas formas tanto en lo que se refiere a cumplimiento de objetivos como en cuanto a coste computacional.

III. CONTROL DE ADMISION EN REDES CELULARES

El control de Admisión es un mecanismo mediante el cual una red decide qué nuevas peticiones de acceso son aceptadas y cuáles no, en función de diferentes criterios. Existen diferentes puntos de vista desde los que se pueden clasificar las políticas de control de acceso: según el tipo de información en la que la decisión se basa (predictiva o no), según su estructura general (familias *product-form*, *trunk reservation* o *general stationary policies*) y según el método que se emplea para configurar cada política (estáticas o adaptativas).

Teniendo en cuenta el primer punto de vista, la mayoría de políticas toman decisiones en función del estado local de la celda en la que operan. La decisión puede tomarse en función de las sesiones activas en la celda o por la cantidad de recursos en uso, etc. De todas formas, la decisión se basa en información de estado local. A parte existen otras políticas que aprovechan la información que se puede obtener de la red para predecir de algún modo eventos futuros, tales como los *handovers*. Esta información predictiva puede obtenerse de las celdas vecinas, de la velocidad de la trayectoria de un móvil, de la forma de las calles y caminos, de la historia reciente del sistema...

Existen numerosos estudios que predicen el comportamiento del sistema y utilizan esta información para tomar decisiones y aceptar o rechazar las peticiones de llamada nuevas o de *handover*. Algunos de estos trabajos son [3] donde el esquema predictivo se basa en información de estado de la celda y sus vecinas, o [4,5] donde se utiliza un esquema predictivo y una política de control de acceso que hace uso de la información obtenida heurística o intuitivamente para mejorar los resultados de esquemas sin predicción.

En este trabajo nos centraremos en políticas de control de admisión de sesiones que utilizan información de estado de la celda local.

Desde el segundo punto de vista, las políticas de control de admisión pueden clasificarse en diferentes familias según sean políticas con reserva de recursos o *trunk reservation*, políticas *product-form* o *general stationary policies*, (ver Fig.1).

Las políticas *product-form* son aquellas que en sistemas multiservicio tienen asociado un proceso de Markov cuya distribución de probabilidades de estados en equilibrio puede escribirse como el producto de las distribuciones marginales de las probabilidades de estado en equilibrio de cada servicio, sujetas a la constante de normalización. Este hecho reduce considerablemente el coste computacional necesario para obtener la distribución de probabilidades de estado en equilibrio. Cuando el número de unidades de recursos no es muy elevado y no hay muchos tipos de servicio diferentes, se puede utilizar el método de la convolución para calcular la distribución de probabilidades de estado en equilibrio [7].

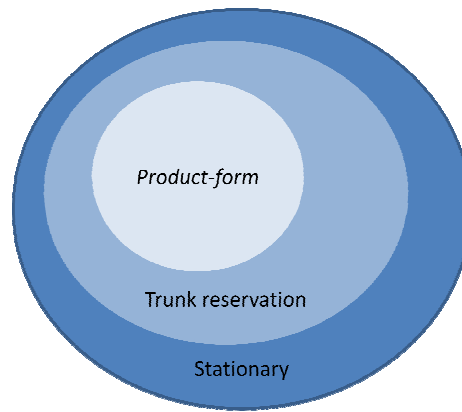


Fig.1. Diferentes familias de políticas de control de acceso

Tradicionalmente las políticas *trunk reservation* son aquellas donde se hace una reserva de recursos de manera explícita. La idea básica es reservar una cierta cantidad de recursos exclusivamente para *handovers* y admitir nuevos flujos de llegada sólo si la cantidad de recursos disponibles para éstos es mayor al número de recursos que precisa la nueva sesión. Uno de los ejemplos más comunes es la política *Fractional Guard Channel* (FGC) [6], en este trabajo estudiaremos más adelante su extensión para escenarios multiservicio.

Las políticas *trunk reservation* dan mejores resultados que las *product-form* en cuanto a capacidad de tráfico, aunque también tienen sus inconvenientes. La máxima capacidad se obtiene si los parámetros están bien ajustados y en el caso de las políticas *trunk reservation* este ajuste puede requerir un coste computacional más elevado que en el caso de las políticas *product-form*.

Se han hecho estudios comparativos entre las dos familias de políticas, para los escenarios estudiados en [8] se muestra que la política *trunk reservation* efectivamente tiene mayor sensibilidad a la configuración de los parámetros, y obtiene mejoras de la capacidad de 5 al 15%. En [9] se estudian escenarios donde los beneficios en cuanto a capacidad de las políticas *trunk reservation* son mayores cuando hay un tipo de servicio con tasa de llegadas baja que necesita muchas unidades de recurso por sesión y probabilidades de bloqueo bajas.

Las políticas *trunk reservation* en general dan buenos resultados pero para algunos escenarios no son óptimas, [10]. En estos casos las políticas de admisión óptimas pertenecen a una familia de políticas más amplia, las *stationary policies* donde la decisión de admisión depende del estado actual del sistema, el cual viene dado por el número de sesiones activas. O en el caso de las *randomized stationary policies*, en las que la decisión depende del estado actual y de una componente aleatoria, [11].

El tercer punto de vista desde el que se pueden clasificar las políticas de control es según el método que se utiliza para configurarlas. Se pueden utilizar procesos de decisión de Markov, semi-markovianos o alternatively se puede utilizar cualquier método de optimización de propósito general. Otro aspecto a tener en cuenta es si la política es estática o adaptativa.

Las políticas estáticas consideran que los parámetros del sistema, tales como las tasas de llegada de peticiones nuevas y de *handover*, son estáticos y diseñan la política de control de acceso para el peor caso. Las políticas adaptativas consideran que estos parámetros no son estáticos y utilizan el historial de los patrones de tráfico, o los estiman periódicamente. En [12] encontramos un ejemplo de esquema adaptativo donde se consideran aplicaciones multimedia con tasas adaptativas.

Como se ha visto, existen distintas familias de políticas utilizadas para el control de admisión. En este trabajo nos centraremos en políticas estáticas, de la familia *trunk reservation* y que utilizan información de estado local sin predicción.

III.1. Descripción del modelo

Nuestro modelo del sistema se centra en lo que ocurre en una sola celda. En ella contamos con un número de unidades de recursos al que llamamos C . El significado físico de estas unidades de recurso dependerá de la tecnología utilizada para implementar el interfaz radio.

Dentro de este sistema contamos con N clases de servicio diferentes que pueden ser gestionados por la celda. Cada uno de estos servicios se distingue por el número de unidades de recurso que utiliza y por los requerimientos en cuanto a QoS (Probabilidades de bloqueo y de terminación forzosa). Además por cada uno de los tipos de servicio se pueden tener nuevos flujos de llegada y flujos de llegadas provenientes de otras celdas o de *handover*. Por tanto, en total se podría decir que hay que servir $2N$ tipos diferentes de flujos de llegada.

En cuanto a la nomenclatura, una nueva llamada con tipo de servicio $i=1, \dots, N$ será la llegada i , mientras que las peticiones de *handover* con tipo de servicio i , serán llegadas $N+i$. Una llamada con tipo de servicio i consume b_i unidades de recurso, donde b_i es un número natural.

Llegados a este punto, pasamos a exponer una serie de aproximaciones que se han considerado en este trabajo con el objetivo de que el modelo del sistema sea tratable matemáticamente. Consideraremos que el proceso de llegadas es de Poisson y que el tiempo de residencia en la celda y la duración de la llamada son variables aleatorias que siguen una distribución exponencial.

Hay otros trabajos que estudian modelos más exactos para este tipo de parámetros, como pueden ser el tiempo de residencia en la celda [13], el *channel holding time* [14,15], el proceso de llegadas [16,17], y el tiempo dentro del área de *handover* [18,19]. Hacer las suposiciones antes mencionadas supone disminuir notablemente la complejidad del modelo y facilitar su análisis, mientras que los resultados que se obtienen son buenas aproximaciones del funcionamiento del sistema. En [20] se estudia el impacto de asumir una distribución exponencial para el tiempo de residencia en la celda, y se llega a la conclusión de que si el tiempo de residencia medio es pequeño en comparación con la duración de las llamadas, suponer que sigue una distribución exponencial es completamente aceptable. Por otro lado, si el tiempo de residencia es mayor, esta suposición se aleja un poco más de la realidad pero se observa que sigue la tendencia general del modelo más

exacto. Del trabajo descrito en [21] también se puede extraer la conclusión de que la distribución exponencial resulta una buena aproximación para el tiempo que transcurre una llamada en el área de *handover*. Y por último lo mismo ocurre con el tiempo entre peticiones de *handover*, tal y como se expone en [22].

Dicho esto, se puede considerar que las suposiciones señaladas anteriormente son aproximaciones perfectamente válidas en nuestro modelo.

Por tanto, nos referiremos como λ_i^n a la tasa de llegadas de los nuevos flujos de llegada con tipo de servicio i y como λ_i^h a la tasa de llegadas de peticiones de *handover* con tipo de servicio i . La duración de las llamadas del tipo de servicio i es exponencialmente distribuida con tasa μ_i^c y el tiempo de residencia en la celda de un tipo de servicio i también es exponencialmente distribuido con tasa μ_i^r . De ahí se tiene que el tiempo que una llamada permanece activa en una celda es exponencialmente distribuido con tasa $\mu = \mu_i^c + \mu_i^r$.

En cuanto a los requisitos de QoS vendrán dados como probabilidades de bloqueo para los nuevos flujos de llegadas y para las peticiones de *handovers* vendrán dadas a partir de probabilidades de terminación forzosa. Definimos un vector $\mathbf{p}=(P_1, P_2, \dots, P_{2N})$ donde las probabilidades de bloqueo de los flujos de llegada nuevos son $P_i^n = P_i$, y la probabilidad de bloqueo en caso de *handover* viene dada por $P_i^h = P_{N+i}$. La relación entre probabilidades de bloqueo en caso de *handover* y terminación forzosa viene dada, tal y como aparece en [23], por:

$$P_i^n = \frac{P_i^h}{\mu_i^c / \mu_i^r + P_i^h} \quad (1)$$

Por otro lado se tiene que en equilibrio estadístico la tasa de peticiones de *handover* está relacionada con la tasa de flujos de llegadas nuevos a través de las probabilidades de bloqueo del siguiente modo, tal y como aparece en [24]:

$$\lambda_i^h = \lambda_i^n \frac{1 - P_i^n}{\mu_i^c / \mu_i^r + P_i^h} \quad (2)$$

El vector de estados multidimensional se define como $\mathbf{x}=(x_1, x_2, \dots, x_N)$, donde x_i es el número de llegadas del tipo i en el sistema sin importar si fueron iniciadas como llegadas nuevas o como llegadas de *handover*.

Finalmente definimos como $b(\mathbf{x})$ al número de unidades de recursos ocupadas en total en el estado \mathbf{x} . La expresión que toma será:

$$b(\mathbf{x}) = \sum_{i=1}^N x_i b_i \quad (3)$$

III.2. Políticas de Control de Admisión estáticas

En la literatura se definen diferentes políticas de control de admisión estáticas. En este apartado se mencionan algunas de ellas y a continuación se señala cual es la opción elegida dentro del modelo descrito anteriormente para conseguir el máximo tráfico que se le puede ofrecer a una red móvil cumpliendo los objetivos de QoS impuestos.

Las principales políticas estáticas de control de admisión de sesión estudiadas en este contexto son las siguientes:

- Complete Sharing (CS). Se admite una petición si hay recursos suficientes para cursarla.
- Integer Limit (IL). Cada servicio tiene asociado dos parámetros, l_i^n y l_i^h números naturales, uno para nuevos flujos de llegada y otro para peticiones de *handover* respectivamente. Una llamada con un tipo de servicio i en el estado \mathbf{x} , será atendida si $x_i^{n,h} + 1 \leq l_i^{n,h}$, donde x_i^n y x_i^h son flujos de llegada nuevos y de *handover* respectivamente del tipo i en el estado \mathbf{x} .
- Fractional Limit (FL). En este caso hay que tener en cuenta cuatro parámetros para un tipo de servicio, t_i^n , t_i^h números naturales, y q_i^n , q_i^h números dentro del intervalo $[0, 1]$. Una llamada con un tipo de servicio i en el estado \mathbf{x} , será atendida si $x_i^{n,h} + 1 \leq t_i^{n,h}$, si no cumple esto los nuevos flujos de llegada serán atendidas con probabilidad q_i^n , y las peticiones de *handover* con probabilidad q_i^h .
- Upper Limit and Guaranteed Minimum (ULGM). Los servicios de tipo i pueden acceder a dos conjuntos de unidades de recursos, uno privado y otro compartido. El número de unidades de recursos reservadas como privadas para los nuevos flujos de llegada con servicio del tipo i será $s_i^n \cdot b_i$, y para peticiones de *handover* será $s_i^h \cdot b_i$, donde las s son números naturales. El resto de unidades de recursos serán compartidas. Una llamada con un tipo de servicio i en el estado \mathbf{x} , será atendida si $x_i^{n,h} + 1 \leq s_i^{n,h}$ o si hay las suficientes unidades de recursos compartidas.
- Multiple Guard Channel (MGC). Cada servicio tiene asociados dos parámetros l_i^n y l_i^h números naturales. Una llamada con un tipo de servicio i en el estado \mathbf{x} , será atendida si $b(\mathbf{x}) + b_i \leq l_i^{n,h}$.
- Multiple Fractional Guard Channel (MFGC). En este caso hay que tener en cuenta cuatro parámetros para un tipo de servicio, t_i^n , t_i^h números naturales, y q_i^n , q_i^h números dentro del intervalo $[0, 1]$. Una llamada con un tipo de servicio i en el estado \mathbf{x} , será atendida si $b(\mathbf{x}) + b_i < t_i^{n,h}$, si por el contrario se cumple $b(\mathbf{x}) + b_i = t_i^{n,h}$ los nuevos flujos de llegada serán atendidas con probabilidad q_i^n , y las peticiones de *handover* con probabilidad q_i^h . Si no se cumple ninguno de los dos casos la llamada es rechazada.

- Randomized Stationary (RS). A cada estado del sistema se le asigna un vector de probabilidades, $q(\mathbf{x}) = \{q_1^n, q_1^h, q_2^n, q_2^h, \dots, q_N^n, q_N^h\}$, una petición con tipo de servicio i que llega en el estado \mathbf{x} será aceptada con probabilidad $q_i^{n,h}(\mathbf{x})$.

Para analizar los resultados de las políticas CS, IL, FL, ULGM, MGC, MFGC (y las combinaciones permitidas), se obtienen las probabilidades de estado, a partir del valor de los parámetros del sistema y de configuración de la política, resolviendo las ecuaciones de balance globales del proceso de Markov bajo estudio, y una vez calculadas éstas se obtienen los parámetros de QoS buscados, en este caso las probabilidades de bloqueo y de terminación forzosa. En el caso de las políticas CS, IL, FL y ULGM, el proceso de Markov resultante es reversible y por tanto las probabilidades de estado tienen una solución en forma de producto (políticas *product-form*). Por lo tanto, se puede utilizar el método de la convolución para resolver las probabilidades de estado con mayor eficiencia. En el caso de las políticas MGC y MFGC (políticas *trunk reservation*), el proceso de Markov resultante no es reversible y las ecuaciones de balance globales se resuelven utilizando un algoritmo iterativo, como puede ser el de Gauss-Seidel.

La complicación se presenta cuando hay que atacar el problema desde la perspectiva inversa, es decir, cuando hay que buscar los parámetros de configuración de la política adecuados que ajusten las necesidades de QoS convenientemente. Atacar este problema de optimización en el caso multidimensional resulta ser difícil y con un gran coste computacional, sobre todo en el caso de las políticas que no son del tipo *product-form* (MGC, MFGC). La familia de políticas de control de acceso RS no presenta estos inconvenientes [25].

En el trabajo descrito en [26] se estudian y comparan estas políticas y combinaciones de las IL, FL y ULGM. La multidimensionalidad del problema y el comportamiento no monotónico del sistema bajo estudio hacen que la determinación de la configuración óptima sea difícil y costosa computacionalmente. En cuanto a la capacidad del sistema, parece ser que la familia de políticas de control de admisión *trunk reservation* proporciona mejores resultados que las *product-form*. Por otro lado, la ganancia relativa disminuye conforme aumenta el número de unidades de recurso C . Y además en el caso de políticas *trunk reservation* se necesita mayor precisión a la hora de determinar los parámetros óptimos para configurarlas que en el caso de las *product-form*. En lo referente a la sensibilidad de las políticas a la sobrecarga de tráfico, las *trunk reservation* ofrecen mejores resultados para soportar estas situaciones.

Según el estudio hecho en [26], la política RS es la que ofrece resultados óptimos en cuanto a capacidad del sistema, pero su configuración puede ser demasiado complicada para las mejoras que ofrece respecto a MFGC. La política de Control de Admisión MFGC presenta buenos resultados en cuanto a capacidad del sistema y su configuración no es demasiado compleja. Por esto se ha elegido esta política de control de acceso para su estudio y configuración eficiente.

III.3. *Multiple Fractional Guard Channel (MFGC)*

Esta política de control de admisión es una generalización de la política MGC. Está dentro de la familia de políticas *trunk reservation*, maneja información de estado local (no utiliza predicción) y los parámetros del sistema se consideran estáticos.

Para definirla a partir de ahora utilizaremos sólo dos parámetros fraccionarios por cada tipo de servicio, pero como se verá a continuación la definición hecha en el apartado anterior encaja perfectamente con la utilizada de aquí en adelante.

Para decidir si una llamada del tipo i es aceptada o no, se hace uso de uno de los dos parámetros fraccionarios t_i^n , t_i^h , dentro del intervalo $[0, C]$, dependiendo de si la llamada es nueva o es de *handover*. De este modo la decisión que se toma sigue este criterio:

- Si $b(\mathbf{x}) + b_i \leq \lfloor t_i^{n,h} \rfloor$, se acepta la llamada.
- Si $b(\mathbf{x}) + b_i = \lfloor t_i^{n,h} \rfloor + 1$, se acepta la llamada con probabilidad $t_i^{n,h} - \lfloor t_i^{n,h} \rfloor$.
- Si $b(\mathbf{x}) + b_i > \lfloor t_i^{n,h} \rfloor + 1$, se rechaza la llamada.

Como se puede observar una petición con tipo de servicio i puede disponer en media de un número máximo de unidades de recurso dadas por $t_i^{n,h}$. Para decidir si una petición es aceptada o rechazada se comparan las unidades de recurso que habrá ocupadas si la petición se acepta con el umbral $t_i^{n,h}$.

IV. IMPLEMENTACIÓN DE MFGC

Para estudiar el ajuste óptimo de los parámetros buscados, partiremos del algoritmo propuesto en [1], que se explicará brevemente en el siguiente apartado. Este algoritmo no depende directamente del método con que se calculen las probabilidades de bloqueo, pero es necesario calcularlas y para ello hay que resolver un proceso de nacimiento y muerte multidimensional. Este cálculo puede ser computacionalmente muy costoso y por ello es necesario optimizar al máximo el algoritmo.

En este trabajo se evaluará el impacto de utilizar una aproximación por fórmulas recurrentes propuesta en [2], para resolver dicho proceso multidimensional y encontrar las probabilidades de bloqueo disminuyendo el coste computacional.

Finalmente dependiendo de los resultados obtenidos se mejorará el algoritmo inicial propuesto y se evaluará para distintos escenarios.

IV.1. Búsqueda de parámetros que ofrezcan capacidad óptima.

Nuestro objetivo es obtener el valor de los parámetros necesarios para configurar el MFGC de forma que se maximice la capacidad del sistema y se cumplan los requerimientos de QoS de cada tipo de servicio. Como se ha señalado anteriormente los requisitos para cumplir la QoS vienen dados en forma de probabilidades de bloqueo máximas permitidas para los nuevos flujos de llegada (B_i^n), y en forma de probabilidades de terminación forzosa máximas permitidas en las peticiones de *handover* (B_i^h). La relación entre la probabilidad de bloqueo de peticiones de *handover* y probabilidad de terminación forzosa se da en la expresión (1).

Definimos λ^T como el agregado de la tasa de llegadas, es decir:

$$\lambda^T = \sum_{1 \leq i \leq N} \lambda_i^n \quad (4)$$

Por otro lado definimos f_i , como la fracción de λ^T que corresponde al servicio i , es decir, $\lambda_i^n = f_i \lambda^T$, teniendo en cuenta que:

$$0 \leq f_i < 1, \quad \sum_{1 \leq i \leq N} f_i = 1 \quad (5)$$

Los parámetros que buscamos estarán definidos en un vector $\mathbf{t}=(t_1, t_2, \dots, t_{2N})$, donde los parámetros t de los flujos de llegada nuevos son $t_i^n = t_i$, y los parámetros t en caso de peticiones de *handover* vienen dados por $t_i^h = t_{N+i}$.

Una vez definidos todos los conceptos, el problema se puede plantear del siguiente modo. Dados el número de recursos del sistema C , las unidades de recurso que ocupa cada servicio b_i , la fracción del tráfico agregado de llegada que corresponde a cada tipo de servicio f_i , las tasas de la duración de las llamadas μ_i^c y del tiempo de residencia μ_i^r , y las probabilidades de bloqueo máxima B_i^n y de terminación forzosa máxima B_i^f , donde $i=1, \dots, N$, se pretende encontrar el valor de los parámetros t , con $2N$ elementos, que definen la política de control de acceso MFGC de forma que se maximice el agregado de tráfico de llegada λ^T cumpliéndose por otro lado que $P_i^n \leq B_i^n$ y $P_i^f \leq B_i^f$, con $i=1, \dots, N$.

Para simplificar la notación definimos un vector $p_{max}=(B_1, B_2, \dots, B_{2N})$, donde las probabilidades de bloqueo máximas de los flujos de llegada nuevos son $B_i^n = B_i$, y las probabilidades de bloqueo en caso de peticiones de *handover* vienen dados por $B_i^h = B_{N+i}$, con $i=1, \dots, N$. A partir de la expresión (1) se deduce fácilmente que:

$$B_i^h = \frac{\mu_i^c}{\mu_i^r} \frac{B_i^f}{1 - B_i^f} \quad (6)$$

El algoritmo propuesto en [1], en el que se basa este trabajo, consta de tres partes diferenciadas. Por un lado una parte principal que se encarga de encontrar el valor máximo buscado del agregado del tráfico de llegadas λ_{max}^T . Por otro lado una segunda parte que para una λ^T dada, decide si es posible alcanzar los objetivos marcados por B_i^n y B_i^f , y al mismo tiempo calcula los valores adecuados de t . Una última parte es la encargada de resolver el proceso de nacimiento y muerte multidimensional, y de este modo obtener las probabilidades de bloqueo del sistema.

La parte principal (ver algoritmo 1) realiza la búsqueda de λ_{max}^T partiendo de un valor inicial, decide si con ese valor inicial se pueden alcanzar los objetivos, si es así va aumentando este valor hasta que ya no se puedan alcanzar. Si por el contrario con el valor inicial los objetivos son inalcanzables, va disminuyéndolo hasta que se encuentra un valor con el que se puedan alcanzar. De este modo, se define un intervalo en el que seguro va a estar el valor buscado de λ_{max}^T . A partir de aquí se va buscando dentro del intervalo, reduciéndolo por uno de los dos extremos a cada paso dependiendo de si el valor con el que está trabajando el algoritmo en cada momento permite alcanzar o no los objetivos. Si permite alcanzarlos se aumenta el límite inferior del intervalo, si no lo permite se disminuye el límite superior. La búsqueda acabará cuando la diferencia entre el valor actual con el que va a trabajar el algoritmo y el valor con el que trabajó en el último paso alcanza un valor dado muy pequeño.

Para determinar si un valor dado de λ^T permite alcanzar los objetivos impuestos se utiliza la segunda parte (ver algoritmo 2), que será invocada en la parte principal.

Algoritmo 1 $(\lambda_{max}^T, t_{opt}) = \text{capacity}(p_{max}, f, \mu_c, \mu_r, b, C)$

```

 $\varepsilon_1 := \langle \text{precisión} \rangle$ 
 $L := 0$ 
 $U := \langle \text{valor inicial} \rangle$ 
(possible,  $t$ ) := solve MFGC( $p_{max}, U, f, \mu_c, \mu_r, b, C$ )
atLeastOnce := FALSE;
if (possible) then
  while possible do
     $L := U$ 
     $t_L := t$ 
    atLeastOnce := TRUE
     $U := 2U$ 
    (possible,  $t$ ) := solve MFGC( $p_{max}, U, f, \mu_c, \mu_r, b, C$ )
  end while {asegura que  $U > \lambda_{max}^T$ }
else
  while not possible do
     $U := L$ 
     $t_L := t$ 
    atLeastOnce := TRUE
     $L := L/2$ 
    (possible,  $t$ ) := solve MFGC( $p_{max}, L, f, \mu_c, \mu_r, b, C$ )
  end while {asegura que  $L < \lambda_{max}^T$ }
end if
repeat
   $\lambda := (L + U)/2$ 
  (possible,  $t$ ) := solve MFGC( $p_{max}, \lambda, f, \mu_c, \mu_r, b, C$ )
  if possible then
     $L := \lambda$ 
     $t_L := t$ 
    atLeastOnce := TRUE;
  else
     $U := \lambda$ 
  end if
until  $(U - L)/L \leq \varepsilon_1$  AND atLeastOnce
 $\lambda_{max}^T := L$ 
 $t := t_L$ 

```

Dado un valor de λ^T , esta parte del algoritmo intenta encontrar los parámetros t adecuados, siempre que existan, que hagan cumplir los objetivos (ver algoritmo 2). Si existen dichos valores se lo comunicará a la parte principal y le dará los valores de t encontrados. Si por el contrario no existen estos t , avisará a la parte principal de que no es posible cumplir objetivos con ese λ^T .

Para encontrar dichos parámetros el algoritmo procede de la siguiente manera: Se inicializa el vector t con valores pequeños, a continuación se procede a buscar componente a componente del vector su valor óptimo en caso de que exista, para el valor de λ^T dado. Para ello, en primer lugar el algoritmo se centra en la primera componente de t . Si su probabilidad de bloqueo asociada es menor a la de los objetivos no se hace nada, si por el contrario es mayor, se le da a la componente

de t el valor máximo que puede tomar C , si para este valor no es posible alcanzar las restricciones impuestas por la QoS resulta que el valor de λ^t es demasiado elevado para alcanzar los objetivos y se lo comunica a la parte principal. Si para el valor máximo sí se pueden cumplir las restricciones se continúa con el algoritmo.

Algoritmo 2 (possible,t)=solveMFGC($p_{max}, \lambda_n, \mu_c, \mu_r, b, C$)

```

 $\varepsilon_2$  :=< precisión >
 $\delta$  :=< valores pequeños >
 $t := (\delta, \delta, \dots, \delta)$ 
 $p := \text{MFGC}(t, \lambda_n, \mu_c, \mu_r, b, C)$ 

repeat
  canConverge:=TRUE;
   $i := 1$ ;
repeat
  if  $p(i) > p_{max}(i)$  then
     $t' := t$ ;  $t'(i) := C$ 
     $p' := \text{MFGC}(t', \lambda_n, \mu_c, \mu_r, b, C)$ 

    if  $p'(i) > p_{max}(i)$  then
      canConverge:=FALSE;
    else
       $L := t(i)$ ;  $U := C$ 
      repeat
         $t(i) := (L + U)/2$ 
         $p := \text{MFGC}(t, \lambda_n, \mu_c, \mu_r, b, C)$ 
        if  $p(i) > p_{max}(i)$  then
           $L := t(i)$ 
        else
           $U := t(i)$ 
        end if
      until  $(1 - \varepsilon_2) \cdot p_{max}(i) \leq p(i) \leq p_{max}(i)$ 
      end if
    end if
     $i := i + 1$ 
  until ( $i > 2N$ ) OR ( NOT(canConverge))

if canConverge then
  if  $p(i) \leq p_{max}(i) \forall i$  then
    possible:=TRUE; exit:=TRUE;
  else
    exit:=FALSE;
  end if
else
  possible:=FALSE; exit:=TRUE;
end if
until exit

```

Dado el valor inicial de la componente de \mathbf{t} , se va aumentando y comprobando el nuevo valor de su probabilidad de bloqueo asociada hasta que ésta esté dentro del margen de valores buscado, $((1-\varepsilon)p_{\max}(i) \leq p(i) \leq p_{\max}(i))$. Cuando ya se ha encontrado el valor de la primera componente del vector \mathbf{t} , se pasa a buscar el valor de la siguiente componente, procediendo del mismo modo. Así sucesivamente para todas las t_i .

Cuando ya se tienen todos los valores, se comprueba si todas las probabilidades de bloqueo cumplen objetivos, si es así se avisa al programa principal de que es posible alcanzar valores de \mathbf{t} adecuados para cumplir objetivos con la λ^T dada y se le dan estos valores. Si por el contrario alguna probabilidad de bloqueo es mayor de lo requerido se vuelve a realizar todo el proceso anterior comprobando cíclicamente cada t_i , hasta que todas las probabilidades de bloqueo sean aceptables.

De este modo se tiene que para λ^T pequeñas la región de convergencia de las t_i es mayor que si se toman valores de λ^T mayores. Si $\lambda^T > \lambda_{\max}^T$ no existe tal región de convergencia y la parte principal procederá a disminuir la λ^T , como se ha explicado anteriormente.

Según lo expuesto, es necesario calcular las probabilidades de bloqueo para cada conjunto de parámetros que se está evaluando y por tanto es necesario resolver el proceso de nacimiento y muerte multidimensional que estamos estudiando. Este proceso se resolverá mediante otro segmento del algoritmo que será invocado cada vez que se necesite calcular probabilidades de bloqueo. Para resolver el proceso multidimensional se deben resolver las ecuaciones de balance global y de este modo obtener la distribución de probabilidades de estado en equilibrio. Llamaremos $p(\mathbf{x})$ a la probabilidad en equilibrio de estar en el estado \mathbf{x} .

Tal y como se propone en [1], se definen en primer lugar los coeficientes $a_i^{n,h}$ que especifican las probabilidades de aceptar una nueva llamada o de *handover* para el tipo de servicio i :

$$a_i^n = \begin{cases} 1 & \text{si } b(\mathbf{x}) + b_i \leq \lfloor t_i \rfloor \\ \lfloor t_i \rfloor - b(\mathbf{x}) - b_i & \text{si } b(\mathbf{x}) + b_i = \lfloor t_i \rfloor + 1 \\ 0 & \text{si } b(\mathbf{x}) + b_i > \lfloor t_i \rfloor + 1 \end{cases} \quad (7)$$

$$a_i^h = \begin{cases} 1 & \text{si } b(\mathbf{x}) + b_i \leq \lfloor t_{N+i} \rfloor \\ \lfloor t_{N+i} \rfloor - b(\mathbf{x}) - b_i & \text{si } b(\mathbf{x}) + b_i = \lfloor t_{N+i} \rfloor + 1 \\ 0 & \text{si } b(\mathbf{x}) + b_i > \lfloor t_{N+i} \rfloor + 1 \end{cases} \quad (8)$$

Si llamamos r_{xy} a la tasa de transición del estado \mathbf{x} al estado \mathbf{y} , se tiene que:

$$r_{xy} = \begin{cases} a_i^n(\mathbf{x})\lambda_i^n + a_i^h(\mathbf{x})\lambda_i^h & \text{si } \mathbf{y} = \mathbf{x} + \mathbf{e}_i \\ x_i\mu_i & \text{si } \mathbf{y} = \mathbf{x} - \mathbf{e}_i \\ 0 & \text{para el resto} \end{cases} \quad (9)$$

Donde \mathbf{e}_i representa un vector cuyas componentes son todas nulas menos la que está en la entrada i . A partir de esto se obtienen las ecuaciones de balance globales:

$$p(\mathbf{x}) \sum_{y \in S} r_{xy} = \sum_{y \in S} r_{yx} p(\mathbf{y}) \quad \forall \mathbf{x} \in S \quad (10)$$

Donde S es el conjunto de estados posibles definido del siguiente modo:

$$S := \left\{ \mathbf{x} : x_i \in \mathbb{N}; \sum_{i=1}^N x_i b_i \leq C; x_i b_i \leq \lceil t_i \rceil, 1 \leq i \leq N \right\} \quad (11)$$

Estas ecuaciones junto con la ecuación de normalización $\sum p(\mathbf{x}) = 1$, permiten calcular los valores de $p(\mathbf{x})$, pero para ello es necesario resolver un sistema que puede llegar a ser de un tamaño bastante considerable y representa la parte más costosa computacionalmente de todo el proceso. En este caso se utiliza el método de Gauss-Seidel para resolver dicho sistema de ecuaciones.

A partir de estos valores se obtienen las probabilidades de bloqueo del siguiente modo:

$$P_i = P_i^n = \sum_{\mathbf{x} \in S} (1 - a_i^n(\mathbf{x})) p(\mathbf{x}) \quad (12)$$

$$P_{N+i} = P_i^h = \sum_{\mathbf{x} \in S} (1 - a_i^h(\mathbf{x})) p(\mathbf{x})$$

Si se tiene en cuenta la expresión descrita en (2) para las λ_i^h , vemos que la tasa de llegadas de las peticiones de *handover* depende de las probabilidades de bloqueo y al mismo tiempo las probabilidades de bloqueo dependen de la tasa de las peticiones de *handover*. Este sistema puede resolverse utilizando, por ejemplo, un método de iteración de punto fijo. En nuestro caso utilizaremos una aproximación con la que se logra eliminar el proceso iterativo, y al mismo tiempo los resultados que se obtienen son adecuados.

En lugar de calcular las λ_i^h a partir de (2), las calculamos del siguiente modo:

$$\lambda_i^h = \lambda_i^n \frac{1 - B_i^n}{\mu_i^c / \mu_i^r + B_i^h} \quad (13)$$

Recordemos que las B_i^n y B_i^h son las probabilidades de bloqueo máximas permitidas que vienen impuestas en el mismo planteamiento del problema como requerimientos de QoS, mientras que P_i^n y P_i^h son las probabilidades de bloqueo asociadas a cada λ_i^h .

Las expresiones (2) y (13) son muy parecidas pero tienen una diferencia que permite acelerar el algoritmo. Utilizando (13) se pueden calcular las λ_i^h sin necesidad de conocer las probabilidades de bloqueo asociadas a ellas P_i^n y P_i^h , las cuales al mismo tiempo necesitarían el valor de λ_i^h , por tanto se consigue eliminar esta dependencia y el problema queda más sencillo. Esta técnica es conocida como speed-up.

El algoritmo descrito en [1] y resumido aquí presenta buenos resultados, pero los tiempos de cálculo pueden ser elevados cuando existen muchos tipos de servicio diferentes o cuando el número de unidades de recurso es muy elevado. Por ello a continuación se estudia una posible aproximación que calcula las probabilidades de bloqueo que nos ocupan con menor coste computacional. La cuestión es si esta aproximación es válida en todos los casos y cuál es el impacto de su utilización. En apartados posteriores se evaluará esta aproximación estudiando estos puntos. Para ello se definirán diferentes escenarios más concretos.

IV.2. Aproximación mediante fórmulas recurrentes.

Para poder determinar los parámetros adecuados de configuración de la política de control de acceso MFGC es necesario calcular las probabilidades de bloqueo para cada conjunto de parámetros que necesite el algoritmo descrito anteriormente. Esto implica la resolución de un proceso de nacimiento y muerte multidimensional que hace que el coste computacional del algoritmo crezca considerablemente. Para resolver esta cuestión en la literatura se ha propuesto una aproximación basada en la utilización de fórmulas recurrentes unidimensionales [2].

La idea clave está en la variable aleatoria $l = \mathbf{x} \cdot \mathbf{b}$, es decir, el número total de unidades de recurso ocupadas $b(\mathbf{x})$ definido en (3). Utilizando esta nueva variable pasamos de trabajar con nuestro vector de estados \mathbf{x} multidimensional, a trabajar con una variable de estado l unidimensional.

Definimos para simplificar la notación una nueva variable que define la probabilidad de que un flujo de llegada del tipo $j=1,2,\dots,2N$, (donde de 1 a N serán flujos de llegada nuevos y de $N+1$ a $2N$ serán peticiones de *handover*), sea admitida cuando el número de unidades de recursos ocupados en el momento de la petición de acceso sean $l=0,\dots,C$:

$$\beta_{l,j} = \left\{ \begin{array}{ll} 1 & \text{si } l + b_j \leq \lfloor t_j \rfloor \\ \lfloor t_j \rfloor - \lfloor t_j \rfloor + 1 & \text{si } l + b_j = \lfloor t_j \rfloor + 1 \\ 0 & \text{si } l + b_j > \lfloor t_j \rfloor + 1 \end{array} \right\} \quad (14)$$

Además se ha de tener en cuenta que en este caso las b_j estarían definidas para $j=1, \dots, 2N$, por tanto para las j dentro del intervalo $[1, N]$ se tiene que $b_j = b_i$, mientras que para las j comprendidas en $[N+1, 2N]$ se cumple que $b_j = b_{j-N}$, donde $j-N=i$.

Como se señala en [2] se puede interpretar como un proceso de nacimiento y muerte unidimensional y obtener de esta forma las ecuaciones de balance globales:

$$(\beta_{l-b_i, i} \lambda_i^n + \beta_{l-b_i, i+N} \lambda_i^h) q(l-b_i) = E\{x_i | l\} (\mu_i^c + \mu_i^r) q(l) \quad (15)$$

Donde $E\{x_i | l\}$ representa el número medio de llamadas con servicio del tipo $i=1, \dots, N$ dado un número total de unidades de recurso ocupadas $l=0, \dots, C$. Multiplicando por b_i y sumando se obtiene la expresión recursiva que buscábamos:

$$\sum_{i=1}^N \frac{(\beta_{l-b_i, i} \lambda_i^n + \beta_{l-b_i, i+N} \lambda_i^h)}{\mu_i^c + \mu_i^r} b_i q(l-b_i) = l q(l) \quad (16)$$

Donde $q(x) = 0$ para $x < 0$ y $\sum_{l=0}^C q(l) = 1$.

A partir de aquí se pueden obtener las probabilidades de bloqueo de la siguiente forma:

$$P_i = P_i^n = \left[1 - (t_i^n - \lfloor t_i^n \rfloor) \right] \cdot q(\lfloor t_i^n \rfloor - b_i + 1) + \sum_{r=\lfloor t_i^n \rfloor - b_i + 2}^C q(r) \quad (17)$$

$$P_{N+i} = P_i^h = \left[1 - (t_i^h - \lfloor t_i^h \rfloor) \right] \cdot q(\lfloor t_i^h \rfloor - b_i + 1) + \sum_{r=\lfloor t_i^h \rfloor - b_i + 2}^C q(r)$$

Resulta mucho más sencillo obtener las probabilidades de bloqueo de este modo que con la expresión (12). Como se señaló al principio el algoritmo es independiente del método que se emplee para calcular las probabilidades de bloqueo. Esta aproximación resultaría una gran ventaja para nuestro algoritmo en cuanto a coste computacional puesto que se necesita calcular muchas veces probabilidades de bloqueo para diferentes valores de los parámetros t y λ^T . Pero es necesario comprobar si realmente nos ofrece resultados válidos en todos los casos. Para ello evaluamos el algoritmo bajo estudio utilizando dicha aproximación para calcular las probabilidades de bloqueo.

	Escenario A	Escenario B
b	[1,2]	[1,2,4,6]
u^c	[1/180,1/300]	[1/300,1/300,1/300,1/300]
u^r	[1/180,1/300]	[1/300,1/300,1/300,1/300]
B^n	[0.02,0.02]	[0.05,0.04,0.03,0.02]
B^{fi}	[0.002,0.002]	[0.005,0.004,0.003,0.002]
B^h^*	[0.01002,0.00668]	[5.0251e-3,4.0161e-3,3.009e-3,2.004e-3]

Tabla 1: Valores de los parámetros de los escenarios A y B

Consideramos dos escenarios diferentes.

Escenario A: Se tiene un sistema con dos servicios distintos ($N=2$) dados por los parámetros definidos en la tabla 1. Falta por definir la fracción del agregado de la tasa de los flujos que llegan correspondiente al servicio i . En el caso de este escenario será $f=[0.2,0.8]$.

Escenario B: Los parámetros que definen este escenario también se definen en la tabla 1. Como se puede observar en este caso se considera un sistema con cuatro tipos de servicio diferentes ($N=4$). El factor de penetración f en este caso se define como:

$$f_i = \hat{f}_i / \hat{F} \quad (18)$$

$$\hat{f}_i = \phi^{i-1}, \quad \phi = 0.2, \quad \hat{F} = \sum \hat{f}_i$$

Si evaluamos el algoritmo descrito anteriormente calculando las probabilidades de bloqueo de las dos formas señaladas, resolviendo el proceso de nacimiento y muerte multidimensional y utilizando la aproximación por fórmulas recurrentes, en cuanto a coste computacional (MFlops) se obtienen los resultados mostrados en la tabla 2. Las tolerancias utilizadas en las diferentes partes de los algoritmos son de 10^{-2} .

Realmente utilizar la aproximación por fórmulas recurrentes ofrece una mejora en cuanto a coste computacional muy elevada, sobretodo conforme van aumentando el máximo número de unidades de recursos disponibles en el sistema C .

*El valor de B^h se ha calculado a partir de la expresión definida en (6)

C	Escenario A		Escenario B	
	Multidimensional	Fórmulas recurrentes	Multidimensional	Fórmulas recurrentes
5	0.27406	0.26539	--	--
10	3.34445	0.61745	315.97000	1.58255
20	30.31446	1.47536	1.1146e+04	5.26828
50	1011.42957	5.38079	--	23.5987
100	11868.99627	17.3247	--	85.63745

Tabla 2: Coste computacional del algoritmo en los escenarios A y B en Mflops

Para comprobar si los resultados obtenidos con la aproximación son fiables, recalculamos las probabilidades de estado en equilibrio del proceso multidimensional resolviendo las ecuaciones de balance globales definidas en (10) y a continuación calculamos las probabilidades de bloqueo mediante (12). Todo ello utilizando los parámetros de configuración para la política de control de acceso MFGC t y el agregado de la tasa de llegadas λ^T obtenidas con el algoritmo utilizando la aproximación mediante fórmulas recurrentes.

Como se puede observar en la Fig. 1 en el caso de dos tipos diferentes de servicio la aproximación que estamos analizando es conservadora, cumple los requisitos de QoS especificados pero en algunos casos es posible que se estén desaprovechando recursos al no ajustar al máximo las probabilidades de bloqueo obtenidos a las máximas permitidas. Esto se hace más evidente conforme se van aumentando el número máximo de unidades de recursos disponibles en el sistema.

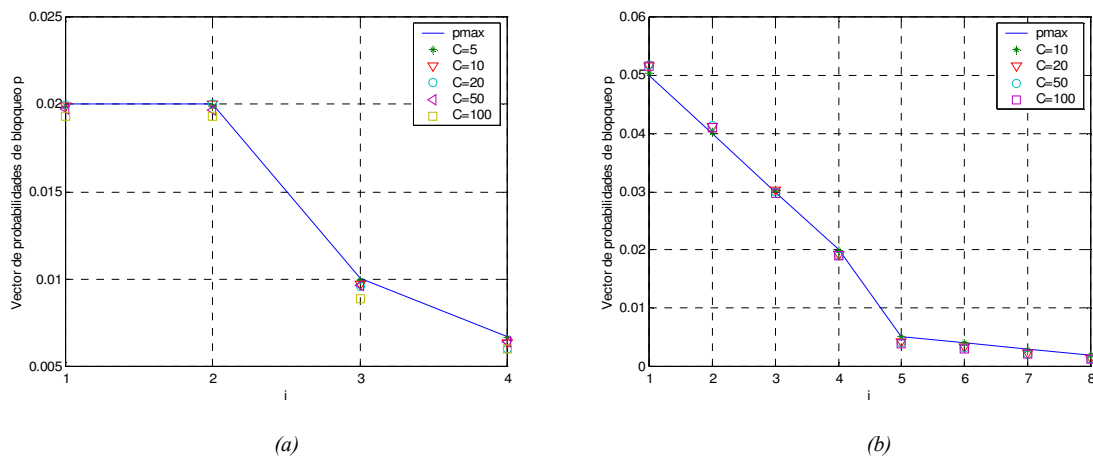


Fig.1. Vector de probabilidades de bloqueo obtenidas a partir de los parámetros calculados mediante la aproximación. (a) Escenario A. (b) Escenario B.

Si nos fijamos en el escenario B, como muestra la fig. 1, vemos que los resultados son bastante peores. Algunas probabilidades de bloqueo superan los umbrales permitidos y están por encima de las probabilidades de bloqueo máximas según los requisitos de QoS.

De todo esto podemos concluir que utilizar la aproximación por fórmulas recurrentes unidimensional para calcular las probabilidades de bloqueo necesarias en el algoritmo bajo estudio no resulta óptimo en todos los casos. Cuando el sistema presenta variedad en cuanto a tipos de servicio, poca uniformidad en las tasas de llegada de cada tipo de servicio y valores de C elevados esta aproximación no ajusta de forma completamente adecuada los parámetros de configuración.

Por ello vamos a proponer un algoritmo que aproveche la rapidez de esta aproximación y los resultados del algoritmo presentado anteriormente que se propone en [1].

IV.3. *Algoritmo final propuesto.*

Tal y como se ha estudiado en el apartado anterior, la aproximación utilizando fórmulas recurrentes unidimensionales no es aceptable en todos los casos. Sin embargo, mejora considerablemente el coste computacional necesario para todo el proceso de búsqueda de los parámetros de configuración de la política MFGC. Por otro lado, utilizar el algoritmo descrito resolviendo las ecuaciones de balance globales del proceso de nacimiento y muerte multidimensional presenta un alto coste computacional, y para sistemas con muchas unidades de recurso y tipos de servicio puede resultar prohibitivo. Por todo ello en este apartado se propone aprovechar las ventajas de la aproximación para mejorar el algoritmo inicial propuesto.

La modificación de este algoritmo consiste básicamente en inicializar los parámetros buscados con los valores obtenidos a partir de la aproximación, es decir, de la utilización de este mismo algoritmo pero calculando las probabilidades de bloqueo a partir de la aproximación por fórmulas recurrentes en los primeros pasos del algoritmo. Una vez obtenidos estos primeros resultados, se utilizan para recalcular estos parámetros utilizando el cálculo de las probabilidades de bloqueo más adecuado. De esta forma se parte de valores más cercanos a los buscados y el algoritmo no hace tantas ejecuciones.

Para encontrar la λ^T óptima, primero se hace un barrido para hallar el intervalo donde se encuentra. En esta mejora del algoritmo el barrido se realiza más finamente puesto que se parte de valores más cercanos al óptimo. En el algoritmo inicial se dobla o divide en dos el valor del agregado de flujos de llegada hasta encontrar el límite superior o inferior de dicho intervalo, en el nuevo planteamiento del algoritmo este valor se aumenta o disminuye un 10% de su valor anterior. Cuando ya se han encontrado los límites superior e inferior del intervalo se va recorriendo y haciendo más pequeño el intervalo de igual forma que se hacía en el algoritmo inicial.

Además para mejorar los tiempos de ejecución también se aprovechan los resultados calculados en ejecuciones anteriores, guardando éstos cuando hayan convergido y utilizándolos en las nuevas

búsquedas. De este modo cada vez que la parte principal del algoritmo llama a su subproceso para averiguar si con el valor de una λ^T dada se pueden alcanzar los objetivos impuestos por la QoS, utiliza como valores iniciales de t los calculados en la llamada anterior a este subproceso siempre y cuando se haya alcanzado convergencia.

Si se desea que los valores de las probabilidades de bloqueo se ajusten lo máximo posible a sus valores máximos, también es necesario controlar que las probabilidades de bloqueo finales que se obtienen con los parámetros λ^T y t calculados estén no sólo por debajo de sus valores máximos permitidos, sino que también estén dentro de un intervalo permitido con un límite inferior definido cercano a la frontera.

Como ahora los valores iniciales tomados para t ya no son valores muy pequeños en todas las ejecuciones de la subrutina mencionada, puede ocurrir que las probabilidades de bloqueo finales asociadas a algunos flujos de llegada de un determinado servicio estén muy por debajo de sus valores máximos. En esta implementación del algoritmo final esto también se tiene en cuenta, ya que se le puede estar dando mayor acceso a los recursos de la celda a flujos de un determinado tipo de servicio que a otros que sería más deseable que los tuvieran.

En el apartado siguiente se estudia en detalle el impacto de esta modificación y las mejoras e inconvenientes que presenta respecto al algoritmo original y a la aproximación como método de cálculo de las probabilidades de bloqueo.

V. COMPARATIVA DE RESULTADOS Y COSTE COMPUTACIONAL

En esta sección se analizan y comparan los resultados obtenidos con el algoritmo bajo estudio y sus variantes. Para ello se comparan el algoritmo original descrito en [1], el caso en que se utiliza la aproximación por fórmulas recurrentes para el cálculo de las probabilidades de bloqueo y el caso en que la aproximación sólo se utiliza para inicializar los parámetros de configuración, es decir, el último algoritmo propuesto. Se comparan los parámetros de configuración de la política MFGC, el cumplimiento de objetivos y finalmente el coste computacional que implica cada método de cálculo de las probabilidades de bloqueo, así como la mejora que se obtiene.

V.1. Escenario con dos tipos de servicio.

Este escenario está definido en la tabla 1, escenario A. Consta de un sistema capaz de cursar dos tipos de servicios distintos. Recordemos que falta por definir la fracción del agregado de la tasa de las peticiones que llegan correspondientes al servicio i . En el caso de este escenario es $f=[0.2,0.8]$. Las tolerancias utilizadas en las diferentes partes de los algoritmos son de 10^{-2} , en el método de Gauss-Seidel de 10^{-4} , menos con $C=50$ y 100 en el que la tolerancia del método de Gauss-Seidel es de 10^{-5} , puesto que se necesita mayor precisión para obtener buenos resultados.

Dentro de este marco se hace un estudio para diferentes valores del número de unidades de recurso disponibles. En este caso consideramos que $C=5, 10, 20, 50$ y 100 .

En primer lugar se comparan los resultados obtenidos para los parámetros de configuración de la política de control de acceso MFGC, definidos anteriormente con el vector t , utilizando las tres variantes del algoritmo bajo estudio. En la figura 2 se muestran las componentes del vector t , en cada punto del eje de abscisas está representada cada una de ellas, recordemos que se corresponderían con $t=[t_1^n, t_2^n, t_1^h, t_2^h]$. Se tienen en el caso de $C=5, 10$ y 20 . En la figura 3 se representan los mismos parámetros pero en este caso cuando $C=50$ y 100 .

Si nos fijamos en la figura 2 y la 3 se observa que en al caso de $C=5$ y $C=10$ y $C=20$ la diferencia entre unos casos y otros es tan apenas apreciable, todos caen aproximadamente en el mismo punto. Conforme se va aumentando el valor de C , los valores de las componentes de t para el caso de utilizar la aproximación por fórmulas recurrentes como único método de cálculo de las probabilidades de bloqueo son algo menores que los valores obtenidos en los otros casos. En cuanto a los valores obtenidos utilizando el algoritmo inicial y el algoritmo final propuesto en este trabajo, vemos que la diferencia es menor para cualquier valor de C . Esto es de esperar puesto que los dos algoritmos en última instancia utilizan el mismo método para el cálculo de probabilidades de bloqueo y las diferencias que pueden surgir son debidas a que no se parte del mismo punto para alcanzar los valores finales.

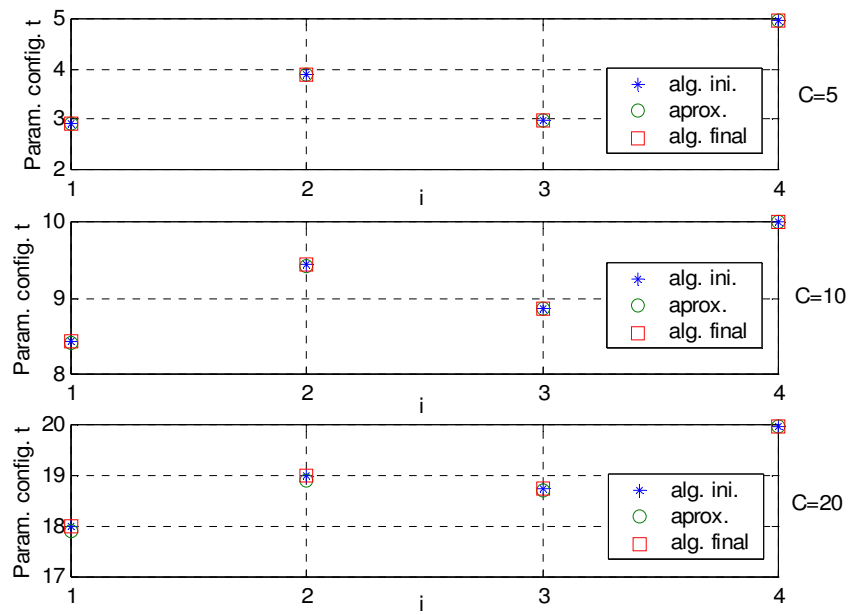


Fig.2. Vector t de parámetros de configuración de la política MFGC para $C=5, 10$ y 20 .

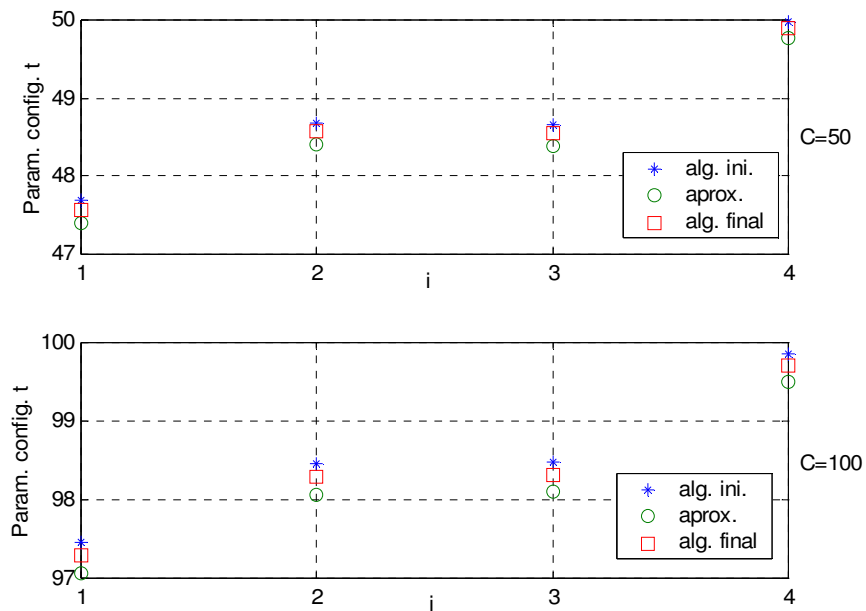


Fig.3. Vector t de parámetros de configuración de la política MFGC para $C=50$ y 100 .

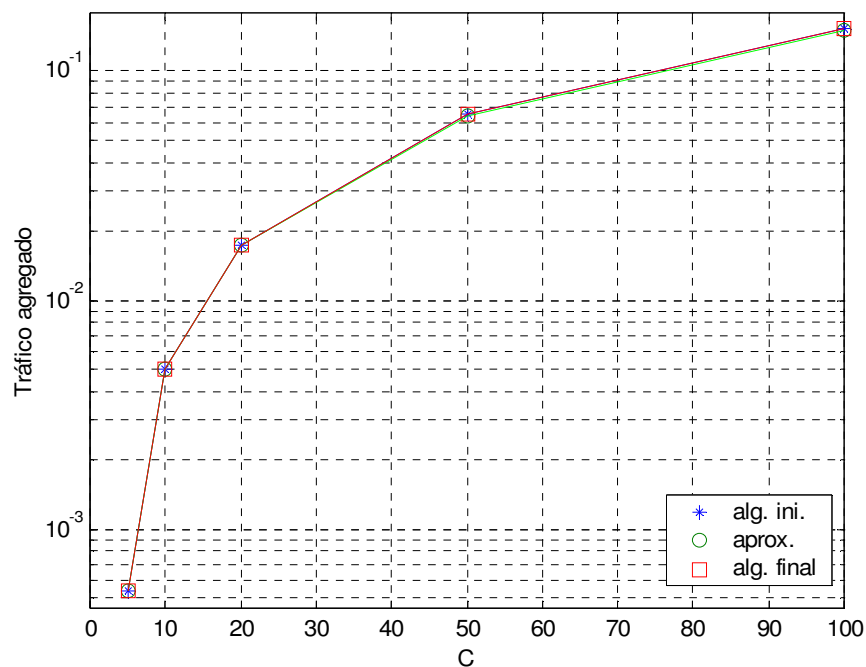


Fig.4. Tráfico agregado λ_{\max}^T frente al número de unidades de recursos disponibles

Además también resulta interesante resaltar que los valores de las componentes t tal y como se observa en las figuras 2 y 3, están marcando un orden de prioridad para los flujos de llegadas. A mayor valor de t_i más prioritarias será la llegada de una petición. Las peticiones de servicio más prioritarias resultan ser las de *handover* con necesidad de más unidades de recurso y las menos prioritarias las nuevas llegadas que necesitan menos unidades de recurso.

En cuanto a los valores alcanzados para el tráfico agregado de flujos de entrada, como se ve en la figura 4, los valores obtenidos son muy parecidos. Para valores pequeños de C la diferencia entre los diferentes métodos utilizados para resolver el problema del cálculo de las probabilidades de bloqueo es tan apenas perceptible, sólo se aprecia una pequeña diferencia para mayores valores de C , donde los valores de tráfico agregado máximo obtenidos mediante el cálculo de las probabilidades de bloqueo por la aproximación mediante fórmulas recurrentes unidimensionales están por debajo de las calculadas con los otros métodos. Al igual que antes, el algoritmo inicial y el final propuesto coinciden prácticamente en los valores del tráfico agregado máximo aceptable por el sistema.

Como se observa en la tabla 3 las probabilidades de bloqueo obtenidas con la aproximación por fórmulas recurrentes no se ajustan tanto a las probabilidades de bloqueo máximas impuestas en los objetivos, sobretodo en el caso de los parámetros correspondientes a peticiones de *handover*.

Las probabilidades de bloqueo reflejadas en la tabla 3 se han calculado resolviendo el proceso de nacimiento y muerte multidimensional para calcular las probabilidades de estado en equilibrio y a partir de éstas obtener dichas probabilidades de bloqueo, para ello en cada caso se han utilizado

los parámetros de configuración finales de la política MFGS y el tráfico agregado máximo obtenidos mediante los tres casos estudiados.

Nos fijamos en las columnas donde viene calculado en tanto por cien el error relativo cometido con respecto a las máximas probabilidades de bloqueo permitidas. Un error positivo denota que las probabilidades de bloqueo obtenidas son menores a las máximas permitidas, es decir, se cumplen los objetivos de QoS que requiere cada tipo de servicio. Por otro lado, un error negativo es lo contrario, significa que las probabilidades de bloqueo están por encima de las máximas y por tanto no se cumplen los objetivos impuestos. Observando la tabla los errores son positivos excepto en un valor de $C=10$ utilizando la aproximación por fórmulas recurrentes.

C	p_{max}	Algoritmo inicial	Error (%)	Fórmulas recurrentes	Error (%)	Algoritmo final	Error (%)
5	$B_1=0.02$	0.0199	0.51143	0.01990	0.51305	0.01983	0.84319
	$B_2=0.02$	0.01987	0.6729	0.01987	0.66140	0.01985	0.74839
	$B_3=0.01002$	0.00999	0.30582	0.00993	0.88999	0.00992	0.95449
	$B_4=0.00668$	0.00665	0.51584	0.00667	0.08597	0.00666	0.26538
10	$B_1=0.02$	0.01986	0.72411	0.01987	0.64920	0.01994	0.28468
	$B_2=0.02$	0.01995	0.22828	0.02000	-0.01076	0.01983	0.84460
	$B_3=0.01002$	0.00993	0.89367	0.00969	3.30137	0.00998	0.37177
	$B_4=0.00668$	0.00663	0.73951	0.00633	5.24791	0.00664	0.6106
20	$B_1=0.02$	0.0199	0.48535	0.01996	0.17713	0.01991	0.42729
	$B_2=0.02$	0.01986	0.69773	0.01999	0.02555	0.01989	0.54845
	$B_3=0.01002$	0.01001	0.10518	0.00958	4.38147	0.00994	0.81056
	$B_4=0.00668$	0.00663	0.72258	0.00602	9.86804	0.00662	0.92246
50	$B_1=0.02$	0.01992	0.41659	0.01970	1.48747	0.01993	0.41659
	$B_2=0.02$	0.01996	0.17886	0.01964	1.80324	0.01993	0.17886
	$B_3=0.01002$	0.00994	0.75970	0.00962	3.98589	0.00993	0.75970
	$B_4=0.00668$	0.00664	0.62619	0.00641	4.00353	0.00667	0.62619
100	$B_1=0.02$	0.01980	0.99642	0.01932	3.39361	0.01986	0.67798
	$B_2=0.02$	0.01989	0.55367	0.01932	3.38178	0.01992	0.38556
	$B_3=0.01002$	0.00997	0.50411	0.00890	11.19439	0.00996	0.62922
	$B_4=0.00668$	0.00667	0.17633	0.00598	10.41748	0.00663	0.77271

Tabla 3: Probabilidades de bloqueo y sus respectivos errores relativos obtenidas con los tres métodos.

Además vemos que en el caso de utilizar la aproximación por fórmulas recurrentes unidimensionales en el proceso completo cuando se cumplen los objetivos, los errores son superiores al 1%, que sería el error máximo deseable. Es más, si nos fijamos en los casos con C mayores los errores relativos son más elevados (mayores a un 10%) sobre todo en el caso de peticiones de *handover*, con lo que se están desperdiciando los recursos del sistema bajo estudio. Se podría decir que la aproximación por fórmulas recurrentes es más conservadora para las llamadas de *handover*, como consecuencia de esto en algún caso los nuevos flujos de entrada pueden tener probabilidades de bloqueo superiores a la de los objetivos.

Por otro lado, como es de esperar, los algoritmos inicial y final propuesto mantienen los errores dentro del 1% deseable y por tanto ajustan mucho mejor los resultados a los límites superiores de las probabilidades de bloqueo, y de este modo permiten que el sistema pueda cursar un mayor número de peticiones.

Uno de los resultados a tener más en cuenta, a parte del cumplimiento de objetivos, es el coste computacional de cada uno de los tres casos. En la tabla 4 se reflejan las MFlops (millones de operaciones) necesarias para alcanzar los valores de configuración óptimos de la política MFGC deseados.

Como se puede ver en el caso de utilizar la aproximación por fórmulas recurrentes el coste computacional es considerablemente menor al del algoritmo inicial, y esta diferencia aumenta todavía más en cuanto C aumenta. En cuanto a los costes computacionales del algoritmo que se ha propuesto en este trabajo en casi todos los casos son mayores al caso de utilizar la aproximación por fórmulas recurrentes en todo el algoritmo y al igual que antes las diferencias son mayores también en cuanto C es más elevada. Sólo en el caso de $C=5$ se ve un coste menor del algoritmo final, esto es debido a que en este algoritmo no sólo se inicializa con la aproximación si no que también se optimizan cálculos, pero esta optimización sólo es notable para un número de unidades de recurso pequeño.

C	Algoritmo inicial	Fórmulas recurrentes	Algoritmo final
5	0.27406	0.26539	0.22187
10	3.34445	0.61745	0.94793
20	30.31446	1.47536	6.62781
50	1011.42957	5.38079	158.15925
100	11868.99627	17.32471	1299.42868

Tabla 4: Coste computacional en MFlops

C	5	10	20	50	100
Mejora (%)	19.04241	71.65654	78.13647	84.3628	89.0519

Tabla 5: Mejora relativa que se experimenta utilizando el algoritmo final respecto al inicial.

En cuanto a los costes computacionales del nuevo algoritmo propuesto respecto el inicial, en la tabla 5 se tiene la mejora para cada escenario del tipo A estudiado, es decir, para diferentes unidades de recurso máximas en la celda. Al igual que el error de la tabla 4, la mejora viene especificada como mejora relativa respecto al coste computacional del algoritmo inicial en tanto por cien, es decir, se calcula a partir de la resta de los dos valores dividida entre el coste del algoritmo inicial.

La mejora es notable, sobre todo cuando la C se hace más grande, llegando para $C=100$ a ser próxima al 90%.

Con todo esto, se puede concluir que en este escenario el método que proporciona mejores resultados en cuanto a coste computacional sería el que emplea la aproximación por fórmulas recurrentes en todo el algoritmo, pero puesto que este método es más conservador con las peticiones de *handover* y puede penalizar a las nuevos flujos de entrada asociándoles probabilidades de bloqueo mayores a las máximas permitidas por los objetivos, podríamos pensar como mejor opción el algoritmo final propuesto, que presenta costes un poco mayores pero aceptables y optimiza más adecuadamente los parámetros de configuración de la política MFGC.

V.2. Escenario con cuatro tipos de servicio.

Los parámetros que definen este escenario se definen en la tabla 1, escenario B. Consta de un sistema capaz de cursar cuatro tipos de servicios distintos, ($N=4$). Recordemos que falta por definir la fracción del agregado de la tasa de los flujos que llegan correspondientes al servicio i . El factor de penetración f en este caso está definido en la expresión descrita en (18). Las tolerancias utilizadas en las diferentes partes de los algoritmos son de 10^{-2} y en el método de Gauss-Seidel de 10^{-4} , menos con $C=50$ en el que la tolerancia del método de Gauss-Seidel es de 10^{-5} , puesto que se necesita mayor precisión para obtener buenos resultados.

Dentro de este marco se hace un estudio para diferentes valores del número de unidades de recurso disponibles. En este caso consideramos que $C=10, 20$ y 50 . No se considera el caso $C=5$ puesto que no tiene sentido una infraestructura de una celda con 5 unidades de recurso que sirve a clientes que demandan 4 tipos de servicio distintos, y uno de ellos consume 6 unidades de recurso en una sesión. En cuanto a $C=100$, no se analiza porque el método de Gauss-Seidel en este caso

necesita aumentar la precisión a 10^{-6} y con los algoritmos aquí estudiados el coste computacional es demasiado elevado.

Al igual que en el escenario anterior, primero se comparan los resultados obtenidos para los parámetros de configuración de la política de control de acceso MFGC, definidos con el vector \mathbf{t} , utilizando las tres variantes del algoritmo bajo estudio. En la figura 5 se muestran las componentes del vector \mathbf{t} , en cada punto del eje de abscisas está representada cada una de ellas, recordemos que se corresponderían con $\mathbf{t}=[t_1^n, t_2^n, t_3^n, t_4^n, t_1^h, t_2^h, t_3^h, t_4^h]$.

Si nos fijamos en la figura 5 se observa que en el caso de $C=10$ la diferencia entre unos casos y otros es tan apenas apreciable, todos caen en el mismo punto. Conforme se va aumentando el valor de C , los valores de las componentes de \mathbf{t} para el caso de utilizar la aproximación por fórmulas recurrentes como único método de cálculo de las probabilidades de bloqueo se separa un poco de los valores obtenidos en los otros casos. Además vemos que con la aproximación los valores son algo más pequeños que con los otros dos casos. En cuanto a los otros conjuntos de valores, utilizando el algoritmo inicial y el algoritmo final propuesto en este trabajo, vemos que la diferencia es tan apenas apreciable para $C=10$ y 20 . Esto era de esperar puesto que los dos algoritmos en última instancia utilizan el mismo método para el cálculo de probabilidades de bloqueo y las diferencias que pueden surgir son debidas a que no se parte del mismo punto para alcanzar los valores finales.

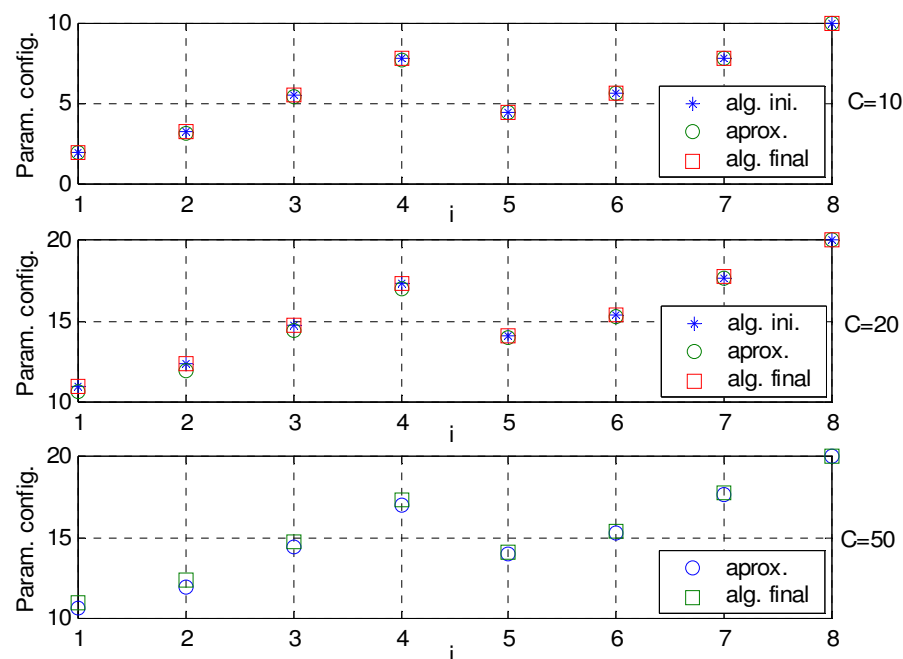


Fig.5. Vector \mathbf{t} de parámetros de configuración de la política MFGC para $C=10, 20$ y 50 .

Además también resulta interesante resaltar que los valores de las componentes t tal y como se observa en las figuras 2 y 3, están marcando un orden de prioridad para los flujos, al igual que en el escenario antes estudiado. A mayor valor de t_i más prioritarias será la llegada de una petición. Por tanto, las peticiones de servicio más prioritarias resultan ser las de *handover* con necesidad de mayor unidades de recurso y las menos prioritarias las nuevas llegadas que necesitan menos unidades de recurso. Además vemos que cuanto mayor es la necesidad de unidades de recurso para un servicio mayor es t y para peticiones con el mismo tipo de servicio vemos que las de *handover* serán más prioritarias.

En cuanto a los valores alcanzados para el tráfico agregado de flujos de llegada entrante, la tendencia es que los valores de tráfico agregado máximo obtenidos mediante el cálculo de las probabilidades de bloqueo por la aproximación mediante fórmulas recurrentes unidimensionales sean menores que los obtenidos con los otros dos casos.

Como se observa en la tabla 6 algunas probabilidades de bloqueo obtenidas con la aproximación por fórmulas recurrentes sobrepasan a las probabilidades de bloqueo máximas permitidas por los requerimientos de QoS, principalmente las correspondientes a nuevos flujos de entrada.

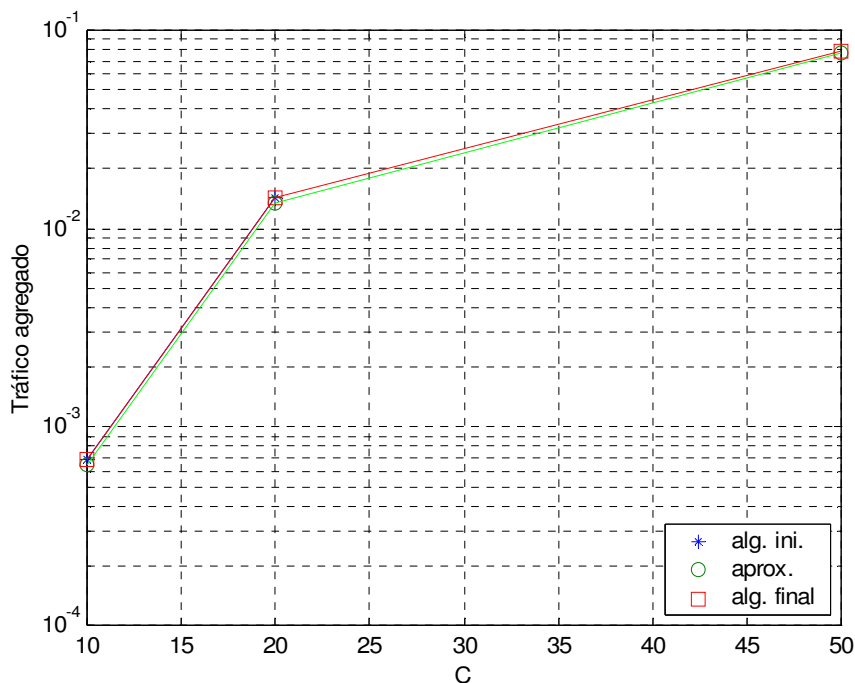


Fig.6. Tráfico agregado λ_{\max}^T frente al número de unidades de recursos disponibles

C	p_{max}	Algoritmo inicial	Error (%)	Fórmulas recurrentes	Error (%)	Algoritmo final	Error (%)
10	$B_1=0.05$	0.04988	0.24363	0.05030	-0.60336	0.04951	0.98401
	$B_2=0.04$	0.03992	0.20382	0.04031	-0.78544	0.03988	0.28890
	$B_3=0.03$	0.02974	0.85593	0.03031	-1.04565	0.02999	0.02108
	$B_4=0.02$	0.01994	0.28351	0.01995	0.23898	0.01989	0.56748
	$B_5=5.025e-3$	5.020e-3	0.09735	5.035e-3	-0.21146	5.012e-3	0.25263
	$B_6=4.016e-3$	4.002e-3	0.3426	3.975e-3	1.03609	3.996e-3	0.51162
	$B_7=3.009e-3$	2.994e-3	0.50977	2.892e-3	3.89608	2.982e-3	0.90045
	$B_8=2.004e-3$	2.0036e-3	0.02012	1.848e-3	7.76598	2.003e-3	0.04776
20	$B_1=0.05$	0.04997	0.05951	0.05159	-3.18010	0.04978	0.44822
	$B_2=0.04$	0.03997	0.07036	0.04118	-2.93826	0.03995	0.13202
	$B_3=0.03$	0.02981	0.62973	0.03021	-0.70685	0.02995	0.15826
	$B_4=0.02$	0.01987	0.62760	0.01926	3.68529	0.0199	0.50014
	$B_5=5.025e-3$	5.020e-3	0.09671	4.260e-3	15.22251	4.998e-3	0.54644
	$B_6=4.016e-3$	4.015e-3	0.01679	3.297e-3	17.88723	4.01e-3	0.13264
	$B_7=3.009e-3$	2.984e-3	0.82122	2.327e-3	22.66979	3.007e-3	0.06427
	$B_8=2.004e-3$	1.997e-3	0.3553	1.376e-3	31.36073	1.99e-3	0.72594
50	$B_1=0.05$	--	--	0.05177	-3.54486	0.04995	0.10341
	$B_2=0.04$	--	--	0.04141	-3.51654	0.03978	0.55375
	$B_3=0.03$	--	--	0.002996	0.13102	0.02980	0.66719
	$B_4=0.02$	--	--	0.01909	4.56965	0.01989	0.55693
	$B_5=5.025e-3$	--	--	4.038e-3	19.63747	5.006e-3	0.38595
	$B_6=4.016e-3$	--	--	3.107e-3	22.64110	4.001e-3	0.37032
	$B_7=3.009e-3$	--	--	2.138e-3	28.94904	2.994e-3	0.51524
	$B_8=2.004e-3$	--	--	1.2e-3	40.1439	1.995e-3	0.44688

Tabla 6: Probabilidades de bloqueo y sus respectivos errores relativos obtenidas con los tres métodos.

Del mismo modo que se hizo anteriormente, las probabilidades de bloqueo reflejadas en la tabla 6 se han calculado resolviendo el proceso de nacimiento y muerte multidimensional para calcular las probabilidades de estado en equilibrio y a partir de éstas obtener dichas probabilidades de bloqueo, para ello en cada caso se han utilizado los parámetros de configuración finales de la política MFGS y el tráfico agregado máximo obtenidos mediante los tres casos estudiados. Recordemos que es necesario resolver las ecuaciones de balance globales, al igual que se hace en el algoritmo inicial y en el propuesto, se resuelven mediante el método de Gauss-Seidel.

Nos fijamos en las columnas donde viene calculado en tanto por cien el error relativo cometido con respecto a las máximas probabilidades de bloqueo permitidas. Un error positivo denota que las probabilidades de bloqueo obtenidas son menores a las máximas permitidas, es decir, se cumplen los objetivos de QoS que requiere cada tipo de servicio. Por otro lado, un error negativo es lo contrario, significa que las probabilidades de bloqueo están por encima de las máximas y por tanto no se cumplen los objetivos impuestos.

Observando la tabla, vemos que usando la aproximación por fórmulas recurrentes unidimensional, aparecen tanto errores positivos como negativos. Si nos fijamos los errores positivos aparecen para valores de llamadas de *handover*. Además vemos que cuando se cumplen los objetivos, los errores son superiores al 1%, que sería el error máximo deseable. Es más, si nos fijamos en los casos con C mayores los errores relativos son más elevados (mayores a un 40%) sobre todo en el caso de peticiones de *handover*, con lo que se están desperdiciando los recursos del sistema bajo estudio. Al igual que en el caso de dos servicios, se podría decir que la aproximación por fórmulas recurrentes es más conservadora para las llamadas de *handover*, como consecuencia de esto en algunos casos los nuevos flujos de entrada pueden estar penalizados y tener probabilidades de bloqueo superiores a la de los objetivos.

Por otro lado, como es de esperar, los algoritmos inicial y final propuesto mantienen los errores dentro del 1% deseable y por tanto ajustan mucho mejor los resultados a los límites superiores de las probabilidades de bloqueo, y de este modo permiten que el sistema pueda cursar un mayor número de peticiones.

C	Algoritmo inicial	Fórmulas recurrentes	Algoritmo final
10	315.97000	1.58522	97.84124
20	1.1146e+4	5.26828	2.2079e+3
50	--	23.59873	2.0217e+5

Tabla 7: Coste computacional en MFlops

C	10	20	50
Mejora (%)	69.034644	80.190368	--

Tabla 8: Mejora relativa que se experimenta utilizando el algoritmo final respecto al inicial.

Otra vez, uno de los resultados a tener más en cuenta, a parte del cumplimiento de objetivos, es el coste computacional de cada uno de los tres casos. En la tabla 7 se reflejan las MFlops (millones de operaciones) necesarias para alcanzar los valores de configuración óptimos de la política MFGC deseados.

Como se puede ver en el caso de utilizar la aproximación por fórmulas recurrentes el coste computacional es considerablemente menor al del algoritmo inicial, y esta diferencia aumenta todavía más en cuanto C aumenta. En cuanto a los costes computacionales del algoritmo que se ha propuesto en este trabajo en todos los casos son mayores al caso de utilizar la aproximación por fórmulas recurrentes y las diferencias son mayores también en cuanto C es más elevada.

En cuanto a los costes computacionales del nuevo algoritmo propuesto respecto el inicial, en la tabla 6 se tiene la mejora para cada escenario del tipo B estudiado, es decir, para diferentes unidades de recurso máximas en la celda. Al igual que el error de la tabla 6, la mejora viene especificada como mejora relativa respecto al coste computacional del algoritmo inicial en tanto por cien, es decir, se calcula a partir de la resta de los dos valores dividida entre el coste del algoritmo inicial.

Con todo esto, se puede concluir que en este escenario el método que proporciona mejores resultados en cuanto a coste computacional sería el que emplea la aproximación por fórmulas recurrentes, pero utilizando este método no se cumplen objetivos para todos los tipos de servicio. El algoritmo final propuesto, presenta costes menores que el algoritmo inicialmente propuesto y ofrece parámetros de configuración de la política MFGC más adecuados, que cumplen con los objetivos de QoS. Pero el coste computacional sigue siendo bastante mayor al algoritmo que utiliza exclusivamente la aproximación por fórmulas recurrentes.

VI. CONCLUSIONES

La finalidad de este trabajo era encontrar un método óptimo y eficiente de cálculo de los parámetros necesarios para configurar la política de control de admisión MFGC, utilizada en redes móviles multiservicio, y obtener con ella resultados que maximicen la capacidad soportada en un sistema celular capaz de cubrir las necesidades de diferentes tipos de servicios.

Para ello se ha estudiado el comportamiento en cuanto a coste computacional del algoritmo desarrollado en [1]. Dado que cuando el número de diferentes servicios o el número de unidades de recursos disponibles en la celda aumenta este algoritmo requiere de un gran coste computacional, haciendo que su utilización en algunos casos sea prohibitiva, se han estudiado diferentes alternativas que mejoren estos resultados.

Se ha llegado a la conclusión que utilizar una aproximación por fórmulas recurrentes unidimensionales para calcular las probabilidades de bloqueo, en vez de resolver el sistema de nacimiento y muerte multidimensional que implica un mayor esfuerzo computacional del algoritmo, no resulta totalmente adecuado puesto que en sistemas que deben soportar bastantes tipos de servicios diferentes con tráficos poco uniformes, las probabilidades de bloqueo no cumplen los objetivos impuestos desde el punto de vista de QoS.

De todos modos esta aproximación puede ser un punto de partida para acelerar dicho algoritmo. En este trabajo finalmente se propone un algoritmo que aprovecha el cálculo óptimo de la capacidad en la celda del algoritmo propuesto en [1] y la rapidez de la aproximación, utilizada en la inicialización de los parámetros buscados.

Esta aceleración del algoritmo presenta una mejora notable en cuanto a coste computacional respecto al algoritmo inicial. Aunque si bien es cierto, que cuando se tienen ambas características del sistema, muchos tipos de servicios y un gran número de unidades de recursos disponibles en la celda, el tiempo de cálculo también puede llegar a ser muy elevado en comparación con el tiempo de cálculo necesario en el caso de utilizar la aproximación por fórmulas recurrentes para el cálculo de las probabilidades de bloqueo.

AGRADECIMIENTOS

El presente trabajo ha sido financiado por el Ministerio de Educación y Ciencia a través del proyecto TSI2005-07520-C03-03.

A título personal me gustaría agradecer el trabajo, el apoyo y el trato que me ha dado mi director de tesina, Jorge Martínez, así como agradecerle la confianza que ha depositado en mí. También quiero agradecer su apoyo y trabajo a los demás miembros del grupo GIRBA como Vicent Pla y Vicente Casares y a mis compañeros de laboratorio, Jose Manuel Giménez, María José Doménech y David García, por su apoyo en los primeros días, por su comprensión y por hacer que la introducción en este mundo sea más fácil.

Gracias a todos.

REFERENCIAS

- [1] Vicent Pla, Jorge Martínez, Vicente Casares, “*Efficient Computation of Optimal Capacity in Multiservice Mobile Wireless Networks.*” 2nd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs '04), Craiglands Hotel, Ilkley, West Yorkshire, U.K, 26th - 28th July 2004. Extended version.
- [2] Felipe A. Cruz-Pérez, José Luis Vázquez-Ávila Lauro Ortigosa-Guerrero, “*Recurrent Formulas for the Multiple Fractional Channel Reservation Strategy in Multi-Service Mobile Cellular Networks.*” IEEE Communication letters, vol. 8, no. 10, October 2004
- [3] SiWu, K.Y. MichaelWong, and Bo Li, “*A dynamic call admission policy with precision QoS guarantee using stochastic control for mobile wireless networks*”, IEEE/ACM Transactions on Networking, 10 (2002), p. 257–271.
- [4] D.A. Levine, I.F. Akyildiz, and M. Naghshineh, “*A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept*”, IEEE/ACM Transactions on Networking, 5 (1997), pp. 1–12.
- [5] Ming-Hsing Chiu and M.A. Bassiouni, “*Predictive schemes for handoff prioritization in cellular networks based on mobile positioning*”, IEEE Journal on Selected Areas in Communications, 18 (2000), pp. 510–522.
- [6] R. Ramjee, R. Nagarajan, and D Towsley, “*On optimal call admission control in cellular networks*”, Wireless Networks Journal (WINET), 3 (1997), pp. 29–41.
- [7] Villy Bæk Iversen, “*Teletraffic Engineering Handbook*”, ITU-D SG 2 and ITC, 2002.
- [8] David Garcia, Jorge Martínez, and Vicent Pla, “*Comparative evaluation of admission control policies in cellular multiservice networks*”, in Proceedings of the 16th International Conference on Wireless Communications (Wireless 2004), July 2004, pp. 517–531.
- [9] S.K. Biswas and B. Sengupta, “*Call admissibility for multirate traffic in wireless ATM networks*”, in Proceedings of IEEE INFOCOM, vol. 2, 1997, pp. 649–657.

- [10]-----, "*Optimal admission control policies in multiservice cellular networks*", in Proceedings of the International Network Optimization Conference (INOC), Oct. 2003, pp. 466–471.
- [11] Sheldon M. Ross, "*Applied probability models with optimization applications*", Holden-Day, 1970.
- [12] L. Huang, S. Kumar, and C.-C.J. Kuo, "*Adaptive resource allocation for multimedia qos management in wireless networks*", IEEE Transactions on Vehicular Technology, 53 (2003), pp. 547–558.
- [13] M. M. Zonoozi and P. Dassanayake, "*Usser mobility modeling and characterization of mobility patterns*" IEEE Journal on Selected Areas in Communications, vol. 15, no. 7, pp. 1239-1252, Sept. 1997.
- [14] F. Barceló and J. Jordán, "*Channel holding time distribution in public telephony systems (PAMR and PCS)*," IEEE Transactions on Vehicular Technology, vol. 49, no. 5, pp. 1615–1625, Sept. 2000.
- [15] Y. Fang and I. Chlamtac, "*Teletraffic analysis and mobility modeling of PCS networks*," IEEE Transactionson Communications, vol. 47, no. 5, pp. 1062–1072, July 1999.
- [16] M. Rajaratnam and F. Takawira, "*Handoff traffic characterization in cellular networks under non classical arrivals and service time distributions*," IEEE Transactions on Vehicular Technology, vol. 50, no. 4, pp. 954–970, July 2001.
- [17] E. Chlebus and W. Ludwin, "*Is handoff traffic really poissonian ?*" Proceedings of ICUPC'95, 1995, pp. 348–353.
- [18] M. Ruggieri, F. Graziosi, and F. Santucci, "*Modeling of the handover dwell time in cellular mobile communications systems*," IEEE Transactions on Vehicular Technology, vol. 47, no. 2, pp. 489–498, May 1998.
- [19] V. Pla and V. Casares-Giner, "*Analytical-numerical study of the handoff area sojourn time*," Proceedings of IEEE GLOBECOM, Nov. 2002.
- [20] F. Khan and D. Zeghlache, "*Effect of cell residence time distribution on the performance of cellular mobile networks*," Proceedings of VTC'97. IEEE, 1997, pp. 949 – 953.
- [21] V. Pla and V. Casares-Giner, "*Effect of the handoff area sojourn time distribution on the performance of cellular networks*," Proceedings of IEEE MWCN, Sept. 2002.
- [22] P. V. Orlik and S. S. Rappaport, "*On the handoff arrival process in cellular communications*," Wireless Networks Journal (WINET), vol. 7, no. 2, pp. 147–157, March/April 2001.
- [23] D. Hong and 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, vol. VT-35, no. 3, pp. 77-92, Aug. 1996, sea also: CEAS Technical Report No. 773, June 1, 1999, college of Engineering and Applied Sciences, State University of New York, Stony Brook, NY 11794, USA.
- [24] Y. –B. Lin, S. Mohan, and A. Noerpel, "*Queueing priority channel assignment strategies for PCS hand-off and initial access*," IEEE Transactions on Vehicular Technology, vol. 43, no. 3, pp. 704-712, Aug. 1994.

- [25] Vicent Pla and Vicente Casares_Giner, “*Optimal Admission Control Policies in Multiservice Cellular Networks*,” in Proceedings of the International Networks Optimization Conference (INOC2003), Paris, France, Oct.2003.
- [26] David García, Jorge Martínez, and Vicent Pla, “*Admission control policies in multiservice cellular networks: Optimum configuration and sensitivity*”, Lecture Notes in Computer Science, Wireless Systems and Mobility in Next Generation Internet, Gabriele Kotsis and Otto Spaniol (eds.) (2005), pp. 121–135.