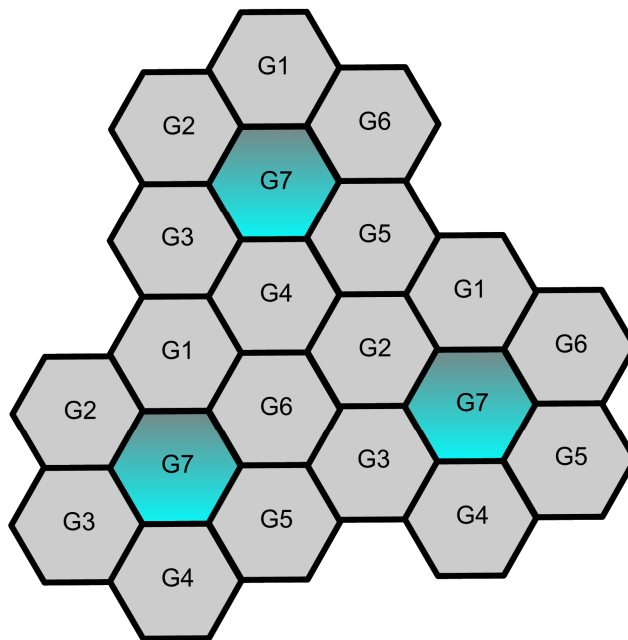


Balance de carga dinámico en redes celulares



Autor: Víctor Molines Villanueva

Director: Jorge Martínez-Bauset

Índice

Índice	1
0. Gráficas, figuras y ecuaciones.	3
1. Introducción, objetivos, antecedentes y contribuciones	11
1.1. Introducción.....	11
1.2. Objetivos y contribuciones.	14
1.3. Antecedentes.....	14
1.4. Metodología.....	14
2. Revisión bibliográfica	17
2.1. El método directo de balance de carga	23
2.1.1. Hybrid Channel Borrowing (HCB).	23
2.1.2. Channel Borrowing Without Locking without channel Rearrangement (CBWL/NR).	24
2.1.3. Directed Retry (DR).	25
2.2. Método indirecto de balance de carga.	27
2.2.1. Channel Borrowing Without Locking with Channel Rearrangement (CBWL/CR).	27
2.2.2. Mobile-Assisted Call Admission algorithm (MACA).	28
2.3. Otras técnicas dinámicas de balance de carga.	30
2.3.1. Hybrid Channel Assignment (HCA).	30
2.3.2. Load Balancing with Selective Borrowing (LBSB).....	30
3. Los métodos de balance de carga en un sistema de dos células	33
3.1. El método directo de balance de carga en un sistema de dos células.....	35
3.1.1. Hybrid Channel Borrowing (HCB).	40
3.1.2. Channel Borrowing Without Locking without channel Rearrangement (CBWL/NR).	40
3.1.3. Directed Retry (DR).	40
3.2. El método indirecto de balance de carga en un sistema de dos células.....	42
3.2.1. Channel Borrowing Without Locking with Channel Rearrangement (CBWL/CR).	45
3.2.2. Mobile-Assisted Call Admission algorithm (MACA).	46
3.3. Otras técnicas dinámicas de balance de carga en un sistema de dos células.....	46
3.3.1. Hybrid Channel Assignment (HCA).	46
3.3.2. Load Balancing with Selective Borrowing (LBSB).....	50
3.4 Descripción del modelo analítico en un sistema de dos células.....	51
3.4.1. Descripción del modelo analítico en un sistema de dos células para el método directo.	51
3.4.2. Descripción del modelo analítico en un sistema de dos células para el método indirecto.	53
3.5. Descripción del método de simulación en un sistema de dos células.	56
3.6. Movilidad en un sistema de dos células.	58
3.7. Resultados y simulación gráfica en un sistema de dos células.....	61
4. Los métodos de balance de carga en un sistema multicelular	68
4.1. Descripción del modelo analítico en un sistema multicelular.	71
4.1.1. Descripción del modelo analítico en un sistema multicelular para el método directo.	72
4.1.2. Descripción del modelo analítico en un sistema multicelular para el método indirecto.	73

4.1.3. Modelo analítico desarrollado para los métodos de balance de carga en un sistema multicelular.....	75
4.2. Descripción del método de simulación en un sistema multicelular compuesto por 3 anillos.	78
4.3. Movilidad en un sistema multicelular.	83
4.4. Resultados y simulación gráfica en un cluster.	87
4.5. Resultados y simulación gráfica en un sistema multicelular.....	92
5. Conclusiones.....	101
6. Referencias.	103
7. Anexos.	104
7.1. Descripción del método de simulación: El entorno de simulación SMPL.....	104
7.2. Descripción del método de simulación en un sistema de dos células.	106
7.3. Descripción del método de simulación en un cluster.	109
7.4. Descripción del método de simulación en un sistema multicelular compuesto por 3 anillos.	116

0. Gráficas, figuras y ecuaciones.

GRÁFICAS.

CAPÍTULO 3.7.

- Para un sistema de dos células, $C=20$, $B=6$, $p=0.5$ y para 5 simulaciones por punto se han realizado las siguientes representaciones gráficas:
 1. PP_A / A para el método directo sin movilidad y mismo A en ambas células.
 2. PP_A / A para el método indirecto sin movilidad y mismo A en ambas células.
 3. PP_A / A_A para el método directo sin movilidad y para A_B fijo.
 4. PP_A / A_A para el método indirecto sin movilidad y para A_B fijo.
 5. PP_A / μ para el método directo sin movilidad y mismo A en ambas células.
 6. PP_A sin movilidad/ PP_A con movilidad para el método directo y mismo A en ambas células.
 7. PP_A / A para el método directo y mismo A en ambas células, con y sin movilidad.
 8. PP_A / A_A para el método indirecto y para A_B fijo, con y sin movilidad.
 9. PP_A / A para el método directo e indirecto, mismo A en ambas células, con y sin movilidad.

CAPÍTULO 4.5.

- Para un sistema *cluster* de 3 células, para 10 simulaciones por punto se han realizado las siguientes representaciones gráficas:
 10. PP_A / A para el método directo sin movilidad y mismo A en un *cluster* de 3 células, para $C=3$, $B=2$ y $p=0.3$.
 11. PP_C / A para el método directo sin movilidad y mismo A en un *cluster* de 3 células para $C=3$, $B=2$ y $p=0.3$.
 12. PP_A / A para el método indirecto sin movilidad y mismo A en un *cluster* de 3 células, para $C=2$, $B=1$ y $p=0.3$.
 13. PP_B / A para el método indirecto sin movilidad y mismo A en un *cluster* de 3 células para $C=2$, $B=1$ y $p=0.3$.

CAPÍTULO 4.6.

- Para un sistema multicelular, para 10 simulaciones por punto se han realizado las siguientes representaciones gráficas:
 14. PP_A / A para el método directo sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=2$, $B=1$ y $p=0.15$.
 15. PP_B / A para el método directo sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=2$, $B=1$ y $p=0.15$.
 16. PP_A / A para el método indirecto sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=1$, $B=1$ y $p=0.15$.
 17. PP_B / A para el método indirecto sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=1$, $B=1$ y $p=0.15$.

- Para un sistema multicelular, para 3 simulaciones por punto se han hecho las gráficas siguientes:
 18. PP_A / A para los métodos directo, indirecto, con y sin movilidad en un sistema multicelular de tres anillos para $C=10$, $B=2$ y $p=0.15$.
 19. $PP_{A \text{ sin bc}} / PP_{A \text{ con bc}}$ para los métodos directo, indirecto, con y sin movilidad en un sistema multicelular de tres anillos para $C=10$, $B=2$ y $p=0.15$.

CAPÍTULO 5.

- Además hemos representado las siguientes gráficas:
 20. Gráfica ejemplo de la probabilidad de bloqueo en función de la intensidad de tráfico en el método analítico directo e indirecto para un sistema de dos células con $C=3$, $B=2$ y $p=0.5$.
 21. Representación del error cometido para una simulación en el método directo mediante el método analítico y mediante simulación para un sistema de dos células con $C=3$, $B=2$ y $p=0.5$.
 22. Representación del error cometido para una simulación en el método indirecto mediante el método analítico y mediante simulación para un sistema de dos células con $C=3$, $B=2$ y $p=0.5$.

FIGURAS.

- Las figuras representadas en nuestro proyecto son:

CAPÍTULO 1.

1. Arquitectura celular.
2. Movilidad en red celular.
3. Modelo básico de una celular representado como un sistema de pérdidas.

CAPÍTULO 2.

4. Ejemplo de área de servicio compuesto por 3 *clusters* de 7 células cada uno.
5. Distancia de reutilización en un área de servicio compuesta por 3 *clusters* de 7 células.
6. Interferencia cocanal causada por varias BS sobre un receptor radio.
7. Sistema celular compuesto por dos células donde se refleja el área TTR para las técnicas Direct Retry y CBWL.

CAPÍTULO 2.1.

8. Ejemplo de sistema celular que usa el método HCB.
9. Esquema de un sistema celular donde se observa el channel locking que provoca un canal prestado desde la célula 1 a la célula 2.
10. Sistema celular compuesto por dos células donde se refleja la situación de dos nuevas llamadas dentro y fuera del área de potencia reducida o TTR y una llamada en curso dentro de la misma para la técnica CBWL.
11. Sistema celular compuesto por dos células para la técnica DR donde se refleja la situación de tres llamadas en la célula B: dos nuevas llamadas (una dentro de la TTR y otra fuera de ella) y una llamada en curso dentro de la TTR. Las llamadas dentro de la TTR son servidas por la célula A.

CAPÍTULO 2.2.

12. Sistema celular compuesto por dos células para el método indirecto donde se refleja la situación de tres llamadas en la célula B: dos nuevas llamadas (una dentro de la TTR y otra fuera de ella) y una llamada en curso dentro de la TTR. Si la llamada Y no pudiera usar un canal debido a su localización, podría intercambiar su canal con Z.
13. Sistema celular compuesto por dos *hot cells*, varios usuarios y agentes para la técnica MACA.

CAPÍTULO 2.3.

14. Esquema de clasificación de usuarios en la técnica LBSB.
15. Algoritmo de asignación de canales en la técnica LBSB.

CAPÍTULO 3.1.

16. Ejemplo de sistema de dos células que usa el método directo con TTR en la zona fronteriza.
17. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=6$ llamadas en curso en la célula A. Posteriormente, se inicia una llamada X1 en TTR y otra X2 fuera de TTR. En ambos casos se asigna una canal a cada llamada que se inicia puesto que quedan canales libres de la célula A que se pueden asignar, resultando $i_A=8$.
18. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X1 en TTR, resultando $i_A=14$, de las cuales 4 son servidas por la célula B.
19. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X2 fuera de la TTR. La llamada no puede ser servida y queda bloqueada.
20. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A =13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B (la llamada 11, 12 y 13). Posteriormente, se termina la llamada activa que usa el canal 10 de la célula A situada fuera de TTR y se intercambia dicho canal por el canal 8 prestado o servido por la célula B que usaba la llamada 13. De esta manera la llamada que usaba el el canal 8 de la célula B ahora usa el canal 10 propio de la célula A que ha sido liberado, resultando $i_A =12$, de las cuales únicamente 2 son servidas por la célula B.
21. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, termina la llamada 13 de la célula A que utiliza el canal 8 prestado por B, resultando $i_A=12$.
22. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A e $i_B=5$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada X1 en la célula B fuera de la TTR, resultando $i_B=6$.
23. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=16$ llamadas en curso en la célula A e $i_B=3$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada X1 en la célula B fuera de la TTR y otra llamada X2 en la TTR. Las dos llamadas quedan bloqueadas. X1 y X2 se bloquean porque el máximo número de canales que se puede asignar a una célula es $C+B$.
24. Ejemplo de sistema congestionado de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=16$ llamadas en curso en la célula A e $i_B=4$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada X1 en la célula A y otra llamada X2 en la célula B. Las dos llamadas quedan bloqueadas.

CAPÍTULO 3.2.

25. Ejemplo de sistema de dos células que usa el método indirecto con TTR en la zona fronteriza.
26. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=6$ llamadas en curso en la célula A y $j_A=4$ llamadas en curso en la TTR. Posteriormente, se inicia una llamada X2 en TTR y otra X1 fuera de TTR, resultando $i_A=8$ y $j_A=5$.
27. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A y $j_A=6$ llamadas en la TTR de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X1 en TTR, resultando $i_A=14$ y $j_A=7$, de las cuales 4 son servidas por la célula B.
28. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, $j_A=7$ llamadas en la TTR de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X2 fuera de la TTR. Se cambia el canal de una llamada activa en la zona TTR a un canal prestado por B, así se libera un canal de A que puede ser utilizado por la nueva llamada X2. El número de llamadas en curso en la célula A se incrementa a 14.
29. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A y $j_A=3$ llamadas en la TTR de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X2 fuera de la TTR. Como no hay llamadas dentro de la TTR a las que se les ha asignado un canal de la célula A, X2 queda bloqueada.

CAPÍTULO 3.3.

30. Sistema compuesto por dos células que usa el método HCA.
31. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$. Inicialmente $i_A=9$ e $i_B=9$. Posteriormente, se inicia una llamada en la célula A y termina una llamada en la célula B. Los valores ahora son $i_A=10$ e $i_B=8$. En este caso los canales del pool no se han asignado.
32. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$. Inicialmente $i_A=10$ e $i_B=29$. Posteriormente, se inicia una llamada en la célula A y termina una llamada en la célula B. Los valores ahora son $i_A=11$ e $i_B=28$. La llamada X1 usa el canal que quedaba en el *pool*, y la llamada 29 libera el canal que utilizaba (P2) y lo asigna al *pool*.
33. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$. Inicialmente $i_A=10$ e $i_B=30$. Posteriormente, terminan dos llamadas en la célula B, la llamada 30 y la llamada 8. Los valores ahora son $i_A=10$ e $i_B=28$. La llamada 30 libera el canal dinámico que utilizaba y lo asigna al pool, la llamada 29 que usa un canal dinámico, cambia dicho canal por el canal 8 fijo liberado.

CAPÍTULO 3.4.

34. Diagrama de estados en un sistema ejemplo de dos células con $C = 3$ y $B = 2$ en el balance de carga indirecto.
35. Diagrama de estados en un sistema de dos células en el balance de carga indirecto.

CAPÍTULO 3.6.

36. Ejemplo de movilidad en un sistema de dos células.
37. Ejemplo de movilidad en un sistema de dos células cuando el usuario está situado en la zona de load balancing.
38. Ejemplo de movilidad en un sistema de dos células cuando el usuario está situado fuera de la zona de load balancing.
39. Ejemplo de movilidad en un sistema de dos células para determinar cuando finaliza la llamada cuando el usuario estaba situado en la zona de load balancing. En el dibujo de la izquierda, el tiempo restante de llamada fuera de la zona de load balancing es mayor que la duración de residencia en esta zona, por lo que la llamada continúa desplazándose. En el dibujo de la derecha, el tiempo restante de llamada fuera de la zona de load balancing es menor que la duración de residencia en esta zona, por lo que se planifica el fin de la llamada para la duración restante de la llamada.
40. Ejemplo de movilidad en un sistema de dos células para determinar cuando finaliza la llamada cuando el usuario estaba situado fuera de la zona de load balancing. En el dibujo de la izquierda, el tiempo restante de llamada en la zona de load balancing es mayor que la duración de residencia en esta zona, por lo que la llamada continúa desplazándose. En el dibujo de la derecha, el tiempo restante de llamada en la zona de load balancing es menor que la duración de residencia en esta zona, por lo que se planifica el fin de la llamada para la duración restante de la llamada.

CAPÍTULO 4.

41. Ejemplo de un sistema multicelular de 3 anillos compuesto por múltiples *clusters*.
42. Ejemplo de un *cluster* de 3 células con TTR en la zona fronteriza.
43. Ejemplo de un *cluster* de 7 células con TTR en la zona fronteriza.

CAPÍTULO 4.1.

44. Ejemplo de un sistema multicelular y los correspondientes *clusters* del sistema.
45. Diagrama de estados para balance de carga directo para un sistema multicelular.
46. Diagrama de estados en un sistema multicelular ejemplo con $C = 3$ y $B = 2$ en el balance de carga indirecto.
47. Sistema de 3 células para el método analítico.

CAPÍTULO 4.3.

48. Ejemplo de movilidad en un sistema multicelular.
49. Ejemplo de movilidad en un sistema multicelular compuesto por zonas de load balancing.
50. Ejemplo de movilidad en un sistema multicelular cuando el usuario está situado fuera de la zona de load balancing.

51. Ejemplo de movilidad en un sistema multicelular cuando el usuario está situado en la zona de load balancing.

CAPÍTULO 4.6.

52. Representación del error en la probabilidad de pérdidas respecto a la célula central para $T_0 = 7 E$ en el sistema multicelular de tres anillos para el método directo sin movilidad y $C=10$, $B=2$ y $p=0.15$.
53. Representación del error en la probabilidad de pérdidas respecto a la célula central para $T_0 = 7 E$ en el sistema multicelular de tres anillos para el método indirecto con movilidad y $C=10$, $B=2$ y $p=0.15$.

ECUACIONES

- Las ecuaciones que hemos utilizado son:

CAPÍTULO 3.4.

1. Ecuaciones de balance de estados en un proceso reversible.
2. Criterio de Kolmogorov.
3. Probabilidad de bloqueo para el método directo en un sistema de dos células.
4. Probabilidad de bloqueo para el método indirecto en un sistema de dos células.

CAPÍTULO 4.1.

5. Probabilidad de bloqueo para el método directo en un sistema multicelular.
6. Probabilidad de bloqueo para el método indirecto en un sistema multicelular.

1. Introducción, objetivos, antecedentes y contribuciones

1.1. Introducción.

Las redes celulares funcionan para proporcionar cobertura radio sobre áreas geométricas llamadas células. El funcionamiento de estas redes requiere la asignación de frecuencias a las estaciones móviles de los usuarios que solicitan los servicios. Sin embargo, las frecuencias de radio son recursos limitados y compartidos, y la administración de estos recursos es una tarea necesaria para el buen funcionamiento de la red.

En cada célula de la red existe una estación base (BS) que proporciona el servicio a través de canales de radiofrecuencia y está localizada en el centro de la célula. La BS se comunica con las estaciones móviles (MS) o usuarios mediante enlaces inalámbricos. Además, un número de células están enlazadas a un Mobile Switching Center (MSC), que actúa como gateway de la red celular con las redes cableadas existentes como Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), cualquier red LAN-WAN o Internet (ver figura 1).

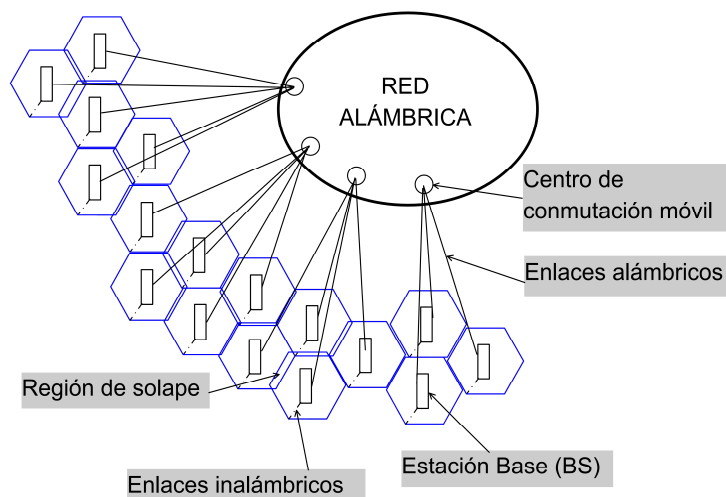


Fig. 1. Arquitectura celular

Los servicios que se prestan en una red celular son comunicaciones o aplicaciones de radio con distintos requisitos de QoS (Quality of Service). Los servicios pueden ser tráfico de voz (llamadas) o tráfico de datos. El tráfico que estudiamos en este proyecto es de streaming o de tiempo real. La red celular que contiene este tráfico garantiza una serie de parámetros temporales y establece un número de canales fijo. Los recursos que se asignan al principio en el servicio de la llamada no varían.

Los usuarios que solicitan los servicios no siempre se encuentran en la misma célula desde su inicio hasta su terminación, sino que se desplazan de una célula a otra. La movilidad del terminal es la capacidad de un terminal inalámbrico de tener acceso a servicios de telecomunicaciones mientras está en movimiento, y también la capacidad de la red para identificar, localizar y seguir ese terminal (ver figura 2).

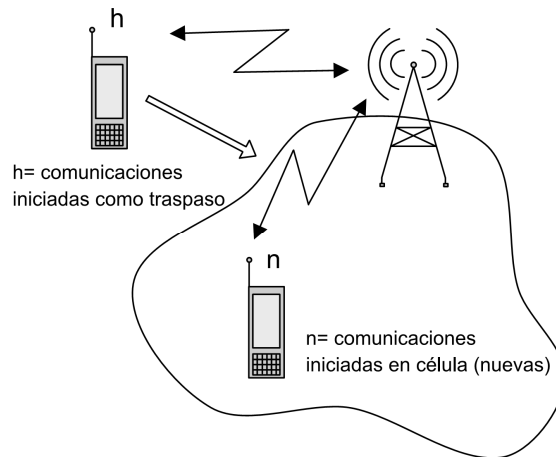


Fig. 2. Movilidad en red celular.

El *handover* es el proceso de transferir una comunicación de una estación base a otra, a medida que el usuario se mueve a través de la red. El manejo de estas transiciones es un factor importante para garantizar la continuidad de las comunicaciones tanto de voz como de datos, caso en el que es muy crítica la pérdida de información. Los usuarios móviles transitan de una célula a otra sin interrupción ni degradación en la calidad de la transmisión. El *handover* hace transparente para el usuario la arquitectura celular que despliega el operador para poder prestar un servicio y soportar la movilidad.

En una red celular es posible tener concentraciones de tráfico elevadas en algunas de las células que componen la red. El tráfico de una red celular fluctúa con el tiempo, es decir, no se mantiene constante. Es posible que se produzcan picos de tráfico, es decir, puntos de congestión de tráfico en el tiempo. Estas células presentan una carga de tráfico demasiado alta para que la propia célula pueda gestionar las nuevas llamadas sin problemas y la probabilidad de bloqueo de llamadas entrantes es mayor comparado con otras células. Son los llamados *hot spots*. El bloqueo que experimentan los *hot spots* también puede afectar a las células vecinas.

En esta situación, el balance de carga de redes celulares ayuda a la gestión eficiente de las solicitudes de llamadas de los usuarios para reducir el bloqueo. De manera general, el balance de carga es un conjunto de técnicas que asignan recursos a las células y cuyo propósito general es mejorar la QoS y la capacidad de la red y evitar problemas como por ejemplo el bloqueo temporal o la terminación forzosa de llamadas. Es importante que un *hot spot* esté bien dimensionado, puesto que el problema del bloqueo no se soluciona únicamente haciendo un balance de carga.

El operador de la red celular se encarga de hacer una planificación de la red celular. En un diseño preliminar de una red celular, el operador dimensiona las células que componen la red en base a estadísticas a priori. La capacidad de la red que diseña depende de la asignación correcta de las frecuencias a los canales de los servicios. Para aumentar la capacidad de una red celular y soportar un mayor número de servicios simultáneos, una misma radiofrecuencia se reutiliza en varias células para comunicaciones distintas. El balance de carga puede ayudar a aliviar el problema de bloqueo cuando el diseño de la red no ha sido el correcto. En este caso será preciso hacer un nuevo diseño para evitar el bloqueo.

El estudio fundamental del proyecto se basa en las técnicas de balance de carga dinámico, que tratan de resolver el problema de los *hot spots* de una forma dinámica, tal que el exceso de tráfico de estas células, también llamadas *hot cells*, será servido por células adyacentes. Mediante dichas técnicas incrementaremos las prestaciones de las células congestionadas sin reducir las de las células vecinas. La idea principal es que las células vecinas puedan prestar canales a células con altas concentraciones de tráfico, si se cumplen unas condiciones de tráfico.

Dichas técnicas las clasificamos en aquellas que usan el método directo y en las que usan el método indirecto. En las primeras, sólo las nuevas peticiones participan en el balance de carga cuando las células han utilizado todos sus canales locales o propios, y en las segundas, todos los usuarios, incluidos los que utilizan llamadas en curso, participan en el balance de carga.

1.2. Objetivos y contribuciones.

A partir del estudio de la literatura, hemos advertido que los estudios realizados para el balance de carga son modelos estáticos y no tienen en cuenta el desplazamiento de los usuarios. Sin embargo, no es una situación realista, porque los usuarios habitualmente se mueven entre células.

Nuestro primer objetivo será el análisis de las prestaciones de los sistemas de balance de carga que tenga en cuenta la movilidad de los usuarios. Nuestro segundo objetivo es hallar la probabilidad de bloqueo como parámetro de mérito de prestaciones, que es la relación entre el número de sesiones nuevas bloqueadas y el número de sesiones nuevas ofrecidas.

Nuestra contribución es la de diseñar y desarrollar un modelo analítico y de simulación que emule un sistema multicelular con movilidad para las técnicas de balance dinámico de carga. Hemos hecho un estudio de la probabilidad de bloqueo en un escenario realista de 3 anillos (37 células) mediante un modelo de simulación de un sistema multicelular real cuando existe movilidad. Para dar una idea del coste temporal de las simulaciones, el tiempo de simulación por punto de nuestro modelo es de 1 hora y media utilizando el entorno de simulación SMPL. Para más información acerca de SMPL ver [9].

1.3. Antecedentes.

Antes de la realización de este proyecto no teníamos ningún conocimiento de simulación de sistemas multicelulares con movilidad del grupo del director del proyecto. De la literatura teníamos sistemas estáticos sencillos como en [1] que nos estimuló a desarrollar soluciones de simulación más complejas.

Dichos estudios se basan en técnicas de asignación de canal sin ahondar en el balance dinámico de carga con movilidad, como en [2], [7]. Las técnicas de balance dinámico más importantes se han estudiado en diversos proyectos mencionados en la bibliografía. También será una parte importante inicial de nuestro proyecto.

1.4. Metodología.

Las técnicas de balance de carga dinámico se pueden estudiar a partir de procesos de Markov. De esta manera evaluamos el comportamiento de los sistemas y calculamos las probabilidades de estado y la probabilidad de bloqueo de los mismos mediante unos programas en Matlab.

Los procesos de Markov obtenidos son de varias dimensiones, según el tipo de técnica analizada y según el tipo de sistema (ya sea un sistema multicelular sencillo o más complejo). La forma de resolverlos es mediante un método recursivo para evitar carga computacional, apoyándonos en el hecho de que los procesos son reversibles. Sin embargo, la solución analítica sólo es posible para sistemas pequeños, pues se requiere mucha carga computacional.

Para contabilizar las llamadas que se inician en cada una de las células de nuestro sistema multicelular, asumimos una población infinita de las fuentes a las cuales se les ofrece servicio con C_b canales. Nuestro modelo básico de una célula es un sistema de pérdidas en el que no se proporciona buffering del tráfico (ver figura 3). Las técnicas de balance de carga parten de este modelo, aunque veremos que presentan características propias. Si una petición no se sirve inmediatamente cuando intenta utilizar un recurso, se aborta. Las peticiones por lo tanto no se hacen cola ($Q=0$). El bloqueo ocurre cuando hay una nueva petición de una fuente, pero todos los servidores están ya ocupados. La probabilidad de pérdidas (probabilidad que servidores y cola estén ocupados) es igual a la probabilidad de bloqueo (probabilidad que los servidores estén ocupados).

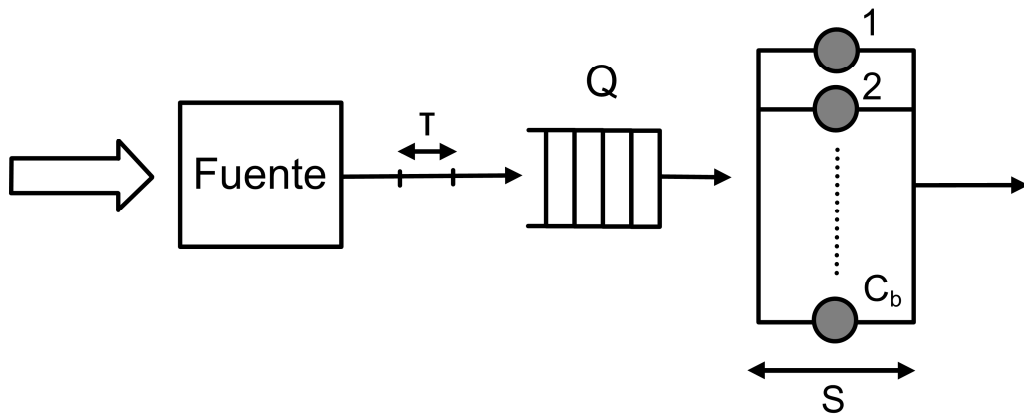


Fig. 3. Modelo básico de una célula representado como un sistema de pérdidas.

Las llegadas de la llamada se pueden modelar por un proceso de Poisson, pero es válido para cualquier distribución estadística de la llamada. Recordemos que un proceso de Poisson expresa la probabilidad de que un número k de eventos ocurran en un tiempo fijo cuando estos eventos ocurren con una frecuencia media conocida y son independientes del tiempo discurrido desde el último evento. La v.a. (variable aleatoria) τ define el tiempo entre llegadas de peticiones de ocupación de los recursos, y la v.a. s define el tiempo de servicio/ocupación de los recursos (channel holding time).

La solución para sistemas más complejos se obtiene mediante simulación. El programa de simulación está realizado en C y se usan unas librerías para gestión de eventos denominadas SMPL. SMPL es un entorno basado en eventos discretos que nos ayuda a representar el comportamiento de nuestros sistemas.

Nuestro propósito es construir un modelo compuesto por una serie de parámetros y reglas de funcionamiento característicos. Los eventos son aquellos sucesos que producen un cambio en el estado de nuestro sistema, por ejemplo, la llegada de un cliente al sistema, el servicio de un cliente o la salida de un cliente.

Mediante la generación y registro de eventos y la generación de datos aleatorios, podremos ser capaces de obtener los resultados de nuestra simulación. La simulación parte primeramente del estudio de un sistema de dos células sin movilidad, luego añadimos la condición de movilidad del usuario, y por último hacemos una abstracción a un sistema multicelular.

Una vez obtenidos los resultados de nuestra simulación y comparados con los resultados del modelo analítico, extraemos conclusiones acerca de las técnicas de balance dinámico estudiadas en los diversos tipos de sistemas. Veremos que los resultados obtenidos a partir de los métodos analíticos son muy parecidos a los obtenidos a partir de la simulación.

2. Revisión bibliográfica

Una de las tareas más importantes que debe realizar un operador es un diseño correcto de una red celular. Entre sus objetivos está aumentar la capacidad de la red celular y soportar un mayor número de llamadas simultáneas. Una de las posibles soluciones es reutilizar una misma radiofrecuencia en varias células para comunicaciones distintas. Sin embargo, en la reutilización de frecuencias puede ocurrir cierto nivel de interferencia en la señal, llamada interferencia cocanal, que es producida por otras células que usan la misma frecuencia, y puede provocar una degradación de la calidad de las comunicaciones (realmente también existe otro tipo de interferencia entre canales adyacentes con menor repercusión).

El estudio de la reutilización de frecuencias se basa en un modelo de células hexagonal. En este modelo existe un número de células suficientemente espaciadas que utilizan el mismo grupo de canales. Es un modelo conceptual que permite un análisis sencillo de un sistema celular. La cobertura de una célula se llama footprint o huella y se puede determinar a partir de los modelos de campo y de propagación.

Podemos definir un área de servicio como aquella zona geográfica donde se proporciona un servicio de comunicaciones. Asumimos que el área de servicio está compuesta por un sistema celular dividido en racimos o *clusters* de células. Cada *cluster* cuenta con un total de S canales duplex disponibles para su utilización. Cada célula tiene asignados un grupo de k canales, donde $k < S$. Si los S canales se dividen entre N células dentro de un grupo o *cluster* donde cada célula tiene el mismo número de canales, el número total de canales radio disponibles se expresa como $S = k \times N$. Un *cluster* se repite M veces dentro del sistema para cubrir el área de servicio. La capacidad del sistema C , será por tanto $C = M \times S$. La capacidad de un sistema se define como el número de llamadas o canales / m^2 . Un ejemplo de un área de servicio se muestra en la figura 4.

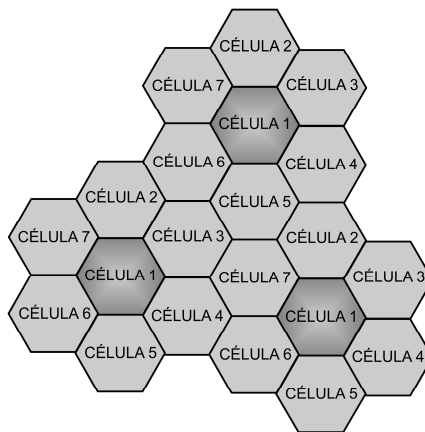


Fig.4. Ejemplo de área de servicio compuesto por 3 *clusters* de 7 células cada uno.

Cuanto más pequeño sea el tamaño del *cluster*, con un tamaño de célula constante, más *clusters* se necesitan para cubrir un área dada y por tanto se consigue mayor capacidad. Cuanto mayor sea N , mayor es la distancia de reutilización de canal, por tanto menor será su interferencia, pero la capacidad del sistema será menor (ver figura 5). El diseñador busca encontrar una solución de compromiso. Existen algunas

formas de incrementar la capacidad del sistema como son la sectorización de las células, el cell splitting, y las aproximaciones de zona de cobertura.

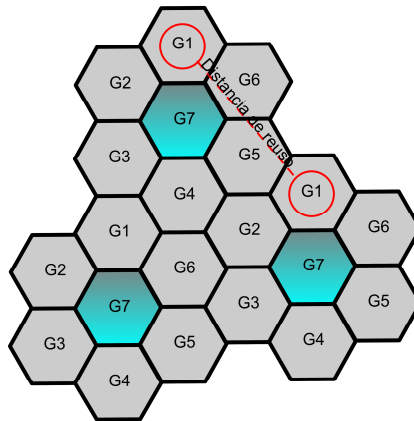


Fig. 5. Distancia de reutilización en un área de servicio compuesta por 3 *clusters* de 7 células.

Un canal puede ser utilizado como cocanal por un usuario de una célula, si la relación carrier-to-interference ratio (CIR) está por encima de un nivel mínimo, es decir, que la potencia recibida desde su propia BS es lo suficientemente grande relacionada con las potencias recibidas de las BS de otras células cuyos usuarios usan ese mismo cocanal (ver figura 6).

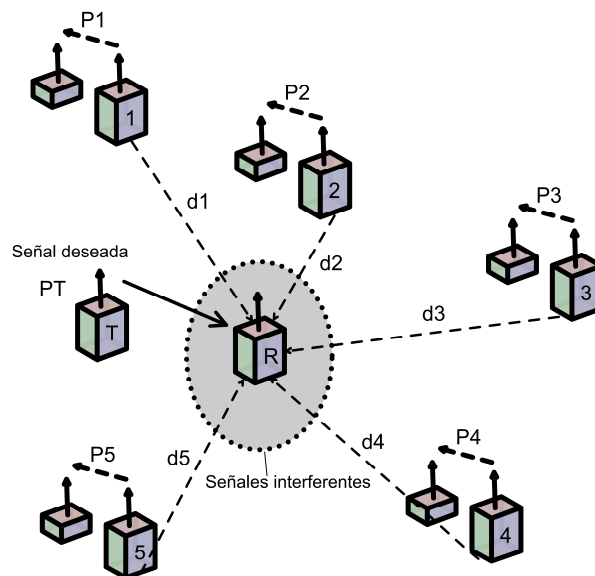


Fig. 6. Interferencia cocanal causada por varias BS sobre un receptor radio.

El procedimiento de asignación de un canal sin interferencia cocanal se basa en asignar correctamente los sets de canales de cada célula. La interferencia cocanal se puede reducir si se aumenta la distancia con las BS vecinas, o si hacemos más pequeñas las potencias interferentes de las estaciones vecinas, o bien si se incrementa la potencia transmitida de la señal deseada manteniendo las potencias de las BS vecinas.

Existen muchas técnicas o estrategias de asignación de canal en las redes celulares [7]. Las podemos clasificar en técnicas fijas (Fixed Channel Allocation o FCA), dinámicas (Dynamic Channel Allocation o DCA) o híbridas (Hybrid Channel Allocation o HCA). La elección de la estrategia de asignación de canales establecerá la

manera en que se gestionan las llamadas cuando un usuario solicita una llamada o cuando un usuario pasa de una célula a otra.

En FCA, las células no utilizan recursos de otras células y se asignan los recursos en base a un algoritmo preestablecido. Existe un número fijo de canales asignados en cada célula. Cuando se origina una llamada dentro de la célula puede ser servida si existen canales libres en la célula. La asignación fija es apropiada para condiciones de alto tráfico de llamadas, pero si el número de llamadas excede al set de canales asignado para la célula, el exceso de llamadas será bloqueado. Además las frecuencias reutilizadas en otras células están lo suficientemente alejadas para que la interferencia cocanal sea despreciable. Este esquema no se adapta a condiciones de tráfico cambiante ni a la distribución de los usuarios en las células. Para solucionar este problema se han propuesto técnicas para pedir prestados canales desde células vecinas menos cargadas que ayudan al balance de carga.

En DCA, los canales se sitúan en un pool de canales y se asignan dinámicamente a una célula según la demanda de tráfico. No hay un número fijo de canales para cada célula como ocurría en la asignación fija. Cuando se hace una petición de llamada, el MSC coloca un canal en la célula que lo solicitó siguiendo un algoritmo que tiene en cuenta varios factores: la frecuencia del canal, la distancia de reutilización y otras funciones de costo. Cuando la llamada termina, el canal vuelve al pool y podrá ser asignado a cualquier célula que lo necesite. En este esquema hay mayor flexibilidad que en la asignación fija de canal y mayor adaptación de tráfico. Sin embargo, las técnicas de asignación dinámicas son menos eficientes que las fijas, en cuanto a probabilidad de bloqueo, para condiciones de alto tráfico de llamadas. HCA es un sistema de balance de carga, mezcla de los dos anteriores, que explicaremos en nuestro trabajo.

Además existen muchos esquemas para implementar estas técnicas de asignación de canal [7], ya sean basados en el área de cobertura de la célula y sus vecinas, o según una CIR mínima medida en los móviles o en las estaciones base, o basados en una asignación de canal centralizada o distribuida.

El operador diseña la red celular para conseguir la máxima capacidad con los mínimos recursos. Sin embargo puede ocurrir que aparezcan *hot spots* si el diseño no ha sido muy riguroso. Las técnicas que vamos a estudiar se basan en el balance dinámico de carga y tratan de resolver el problema de los *hot spots* de una forma dinámica, tal que el exceso de tráfico de las *hot cells* será servido por células adyacentes. Mediante dichas técnicas incrementaremos las prestaciones de las células congestionadas sin reducir las de las células vecinas.

En primer lugar, clasificaremos las técnicas de balance dinámico en técnicas dependientes y en técnicas independientes de la zona en que se localizan los usuarios. En las primeras, la ejecución de dichas técnicas se puede aplicar sólo si los usuarios están en una zona específica, y en las últimas, la ejecución de las mismas no depende de la localización de los usuarios.

En las técnicas de zona específica, sólo los usuarios situados en una porción de área de la célula llamada Transferable Traffic Region (TTR), pueden participar en el balance de carga. La probabilidad de que un usuario pueda participar en él dependerá de la localización y el tamaño de la TTR y de la distribución de los usuarios en la célula.

La región TTR no necesariamente está situada en el borde de las células. Dentro de este tipo de técnicas, existen varias como Channel Borrowing Without Locking (CBWL), cuya TTR está centrada alrededor de la BS y que a su vez se divide en dos, CBWL sin reorganización (CBWL/NR) y con reorganización (CBWL/CR); y Directed Retry (DR) cuya TTR está situada en el borde de la célula (ver figura 7).

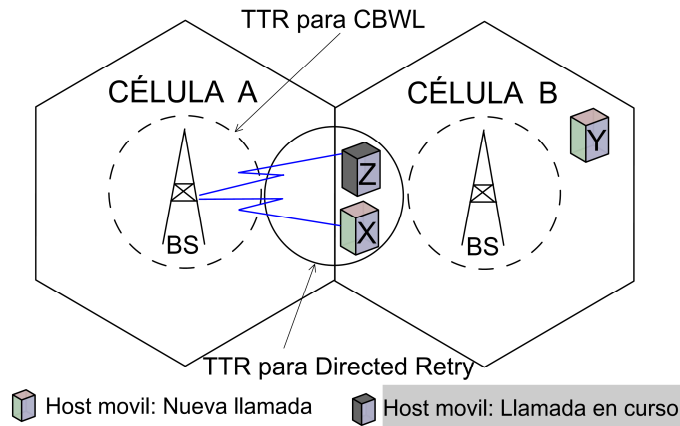


Fig. 7. Sistema celular compuesto por dos células donde se refleja el área TTR para las técnicas Direct Retry y CBWL

En las técnicas de balance dinámico aplicables en cualquier zona de la célula, los usuarios no necesitan estar situados en ninguna localización en concreto para que dichas técnicas se puedan aplicar. Los esquemas o técnicas que forman parte de esta clase son Hybrid Channel Assignment (HCA), Mobile-Assisted Call Admission Algorithm (MACA), Hybrid Channel Borrowing (HCB) y Load Balancing with Selective Borrowing (LBSB). Estas técnicas, al igual que las anteriores de zona específica, las explicaremos en los siguientes capítulos.

A partir de la aquí también podemos hacer una clasificación adicional de los tipos de balance de carga dinámico, que será la que tomemos como referente en nuestro proyecto [1]. En primer lugar, el método directo, implementado por HCB, CBWL/NR y DR. En aquellas células que han utilizado todos sus canales locales, sólo las nuevas peticiones participan en el balance de carga, es decir, sólo estas peticiones pueden usar los canales prestables o pueden ser servidas por células vecinas (forwarded calls).

El indirecto es el otro tipo de balance de carga dinámico y es típico de los esquemas CBWL/CR y MACA. Todos los usuarios, incluidos los que utilizan llamadas en curso, participan en el balance de carga. En los esquemas de zona de tipo indirecto, si un nuevo usuario no puede usar un canal prestable o no puede ser servido por una *cold cell*, entonces una llamada en curso puede cambiar su canal a un canal prestable o ser servido por la otra célula no congestionada, para liberar el canal fijo que usaba y entregárselo a la llamada que pedía servicio.

LBSB y HCA son técnicas de balance dinámico que en su concepción son técnicas directas, pero debido a sus características concretas requieren un estudio y análisis distinto.

A continuación exponemos la clasificación de las técnicas de balance dinámico según dos perspectivas.

En la primera visión, partimos de considerar donde está localizado el usuario.

- Técnicas dependientes de zona: DR y CBWL
 - Directo: CBWL/NR y DR
 - Indirecto: CBWL/CR
- Técnicas independientes de zona: HCB, MACA, HCA y LBSB
 - Directo: HCB
 - Indirecto: MACA
 - Otras técnicas: HCA y LBSB

En la segunda, clasificamos las técnicas partiendo de qué usuarios participan en el balance de carga.

- Técnicas directas: HCB, DR y CBWL/NR
 - Dependientes de zona: DR y CBWL/NR
 - Independientes de zona: HCB
- Técnicas indirectas: CBWL/CR y MACA
 - Dependientes de zona: CBWL/CR
 - Independientes de zona: MACA
- Otras técnicas: HCA y LBSB

El propósito de nuestro proyecto es estudiar varios tipos de técnicas de balance de carga dinámico, y en especial realizar un programa de simulación que emule fielmente el comportamiento de las técnicas de balance directo e indirecto en un sistema multicelular con movilidad. Este programa debe proporcionarnos unos resultados que representan las probabilidades de estado y probabilidades de bloqueo conformes a las técnicas estudiadas.

Para evaluar el comportamiento de estas técnicas se desarrolla un modelo analítico matemático basado en procesos de Markov que se puede obtener de la bibliografía [1]. Con este fin realizamos un programa en Matlab que resuelva las expresiones matemáticas de los modelos analíticos.

Primero realizamos un programa en Matlab que evalúa la probabilidad de bloqueo del método directo en base al proceso de Markov que lo caracteriza y posteriormente hacemos lo mismo para el método indirecto.

Para el análisis de los resultados de nuestro programa de simulación se ha hecho un estudio variando los parámetros principales del programa para cada tipo de técnica de balance y para varios tipos de sistemas partiendo de un modelo de dos células y terminando con un modelo multicelular de 3 anillos.

2.1. El método directo de balance de carga

El método directo [1] es una técnica de balance de carga dinámico en la que sólo las nuevas peticiones de llamadas participan en el balance de carga. Consideramos un grupo de técnicas que utilizan el método directo que explicaremos a continuación. Son HCB, CBWL/NR y DR.

2.1.1. Hybrid Channel Borrowing (HCB).

El Hybrid Channel Borrowing [2] es un esquema dinámico de balance de carga de tipo directo independiente de zona que mejora las prestaciones de la asignación fija de canal FCA. Es un tipo de estrategia para pedir prestados canales a las células adyacentes donde el número de canales en cada set de canales se determina a priori dependiendo de las condiciones históricas de tráfico. Existen más estrategias de borrowing [6], como el Simple borrowing (SB) o el Channel Ordering (CO).

A continuación presentamos en la figura 8 un sistema celular que utiliza el método HCB. A^i, B^i, C^i son el conjunto de canales asignados a cada una de las células que componen el *cluster*. (A, B), (A, C) y (B, C) son conjuntos de canales que pueden ser pedidos por las células que se indican. E_i es el tráfico ofrecido en cada célula. c_s indica el número de sets de canales necesarios para dar servicio (en este caso $c_s=3$). σ es el cociente entre la distancia física entre los centros de las dos células que usan los mismos canales y el radio de una célula.

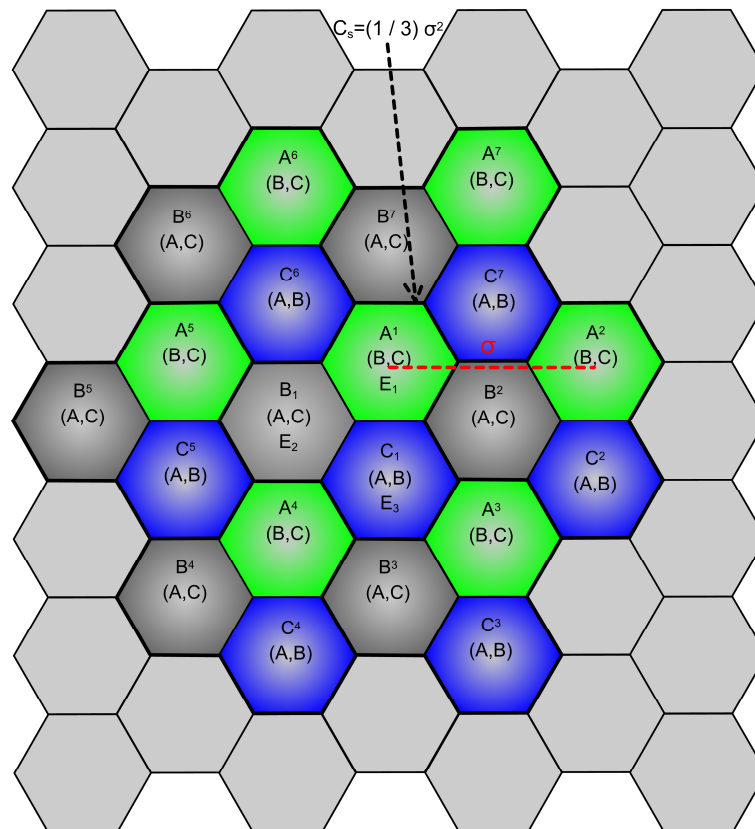


Fig. 8. Ejemplo de sistema celular que usa el método HCB.

En esta técnica, sólo se puede pedir prestado un canal a una célula, si este canal prestado no interfiere con las llamadas existentes. Cuando se presta un canal, otras células adyacentes a la célula a la que se le presta el canal no pueden utilizar este canal ya que se produciría una interferencia cocanal. Es lo que se llama channel locking (ver figura 9). El número de células que no pueden utilizar ese canal depende del layout del sistema y de los canales asignados a cada célula.

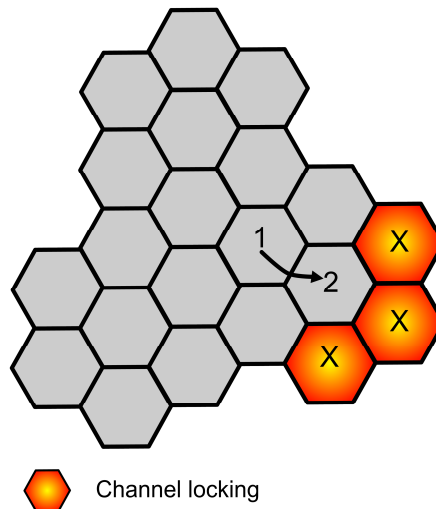


Fig. 9. Esquema de un sistema celular donde se observa el channel locking que provoca un canal prestado desde la célula 1 a la célula 2. Impide que tres células adyacentes a la célula 2 puedan utilizar dicho canal.

2.1.2. Channel Borrowing Without Locking without channel Rearrangement (CBWL/NR).

CBWL [5] es un esquema dinámico de balance de carga de zona específica que también se basa en channel borrowing como HCB. Cuando hay un set de canales agotado, los canales prestados se usan con potencia de transmisión reducida para limitar la interferencia cocanal desde células adyacentes (ver figura 10). No se necesita channel locking, puesto que no hay interferencia cocanal.

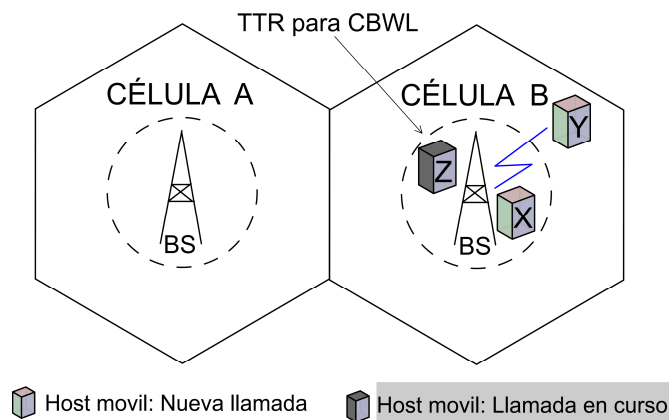


Fig. 10. Sistema celular compuesto por dos células donde se refleja la situación de dos nuevas llamadas dentro y fuera del área de potencia reducida o TTR y una llamada en curso dentro de la misma para la técnica CBWL.

Para saber si una MS está en esa área, la BS transmite una señal con potencia reducida llamada BCSS (Borrowed Channel Sensing Signal), cuya potencia debe estar por encima de un umbral en la MS.

Existen 2 tipos de técnicas CBWL [1]:

- CBWL/NR: CBWL without channel rearrangement, en el que las nuevas peticiones de llamada piden prestados canales dentro de una fracción de área (área de channel borrowing o TTR). Es una técnica del método directo.
- CBWL/CR: CBWL with channel rearrangement, en el cual incluso si una nueva petición de llamada está fuera de una fracción de área (área de channel borrowing), puede dársele servicio, si existe al menos un usuario activo dentro del área. En este caso, el usuario activo puede pedir prestado un canal de su célula vecina con un nivel de potencia reducido y liberar su canal que será asignado a la nueva petición de llamada. Se trata de una técnica del método indirecto.

2.1.3. Directed Retry (DR).

Directed Retry [4] es un esquema dinámico de balance de carga de tipo directo de zona específica en la cual los usuarios escuchan el transmisor de su célula y de las células vecinas. Permite a usuarios del sistema móvil buscar canales de radio libres en más de una célula, por lo que los usuarios pueden usar un canal de las células vecinas.

DR incrementa el número de canales potenciales y la utilización del canal. Solamente los usuarios de la región solapada pueden recibir la llamada de los transmisores de las células vecinas. En DR la carga de tráfico en la región de solape se comparte, no se piden prestados canales (ver figura 11).

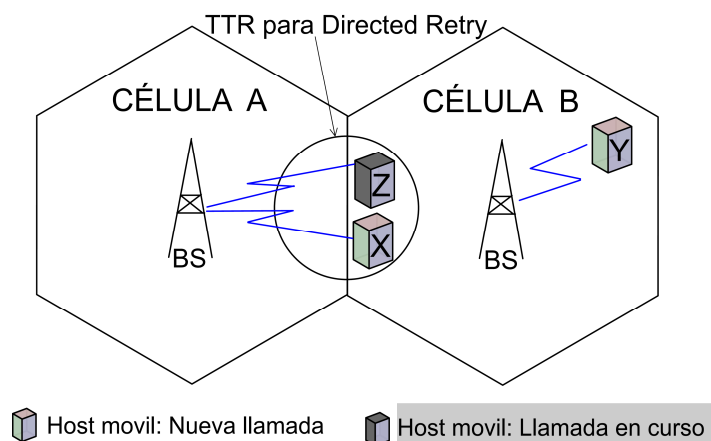


Fig. 11. Sistema celular compuesto por dos células para la técnica DR donde se refleja la situación de tres llamadas en la célula B: dos nuevas llamadas (una dentro de la TTR y otra fuera de ella) y una llamada en curso dentro de la TTR. Las llamadas dentro de la TTR son servidas por la célula A.

El procedimiento llevado a cabo por DR cuando se solicita una llamada es el siguiente:

- La oficina de conmutación (MSC) se encarga de recibir los intentos (attempts) de llamada y si todos los canales están ocupados, avisa a un usuario (el equipo telefónico móvil) para comprobar canales de células vecinas.
- Cada usuario chequea la fuerza de la señal de los transmisores de células vecinas.
- Si la señal tiene suficiente fuerza y si encuentra canal, entonces la llamada es iniciada usando ese canal. Si no, el intento de llamada falla.
- El usuario suele hacer una búsqueda para el canal con una calidad adecuada.
- Si hace más intentos de búsqueda, aumenta la probabilidad de repartir un canal con calidad inadecuada.

2.2. Método indirecto de balance de carga.

El método indirecto [1] es una técnica de balance de carga dinámico en el que todos los usuarios, incluidos los envueltos en una llamada participan en el balance de carga. El método indirecto se puede aplicar tanto a las técnicas dependientes de la localización de la llamada como a las independientes, sin embargo, tiene un uso mayor en las primeras, puesto que para las técnicas aplicables en cualquier zona de la célula, el método directo y el método indirecto son equivalentes.

En el caso de las técnicas empleadas en el balance de carga dinámico de zona específica, si un nuevo usuario no puede usar un canal debido a su localización, entonces una llamada en curso dentro de la TTR puede cambiar su canal a un canal prestable o ser servido por la otra célula no congestionada, para liberar el canal que usaba y entregárselo a la nueva llamada fuera de la zona de TTR que pedía servicio (ver figura 12). La llamada en curso debe utilizar un canal propio de la célula y no un canal que participe en balance de carga, si no el intercambio del canal no se podría realizar. En el caso de las técnicas en que el balance de carga se puede emplear en cualquier zona de la célula, no existe esta restricción de localización y la asignación de un canal de balance de carga es siempre posible.

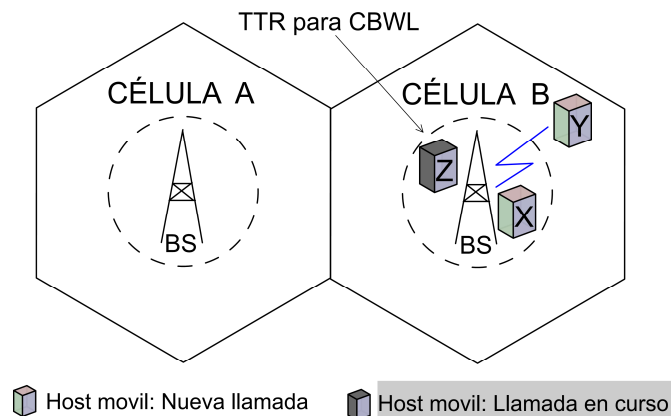


Fig. 12. Sistema celular compuesto por dos células para el método indirecto donde se refleja la situación de tres llamadas en la célula B: dos nuevas llamadas (una dentro de la TTR y otra fuera de ella) y una llamada en curso dentro de la TTR. Si la llamada Y no pudiera usar un canal debido a su localización, podría intercambiar su canal con Z.

A continuación explicamos las técnicas que utilizan el método de balance indirecto: CBWL/CR y MACA.

2.2.1. Channel Borrowing Without Locking with Channel Rearrangement (CBWL/CR).

CBWL/CR [1] se trata de un esquema dinámico de balance de carga de tipo indirecto de zona específica. Es una técnica particular de CBWL en la que podemos dar servicio a una nueva petición de llamada que está fuera de una fracción de área (área de *channel borrowing*), si existe al menos un usuario activo dentro de esta área. El usuario activo puede pedir prestado un canal de su célula vecina con un nivel de potencia reducido y liberar un canal que será asignado a la nueva petición de llamada.

2.2.2. Mobile-Assisted Call Admission algorithm (MACA).

MACA [3] es un esquema dinámico de balance de carga de tipo indirecto independiente de zona. Enunciamos sus características principales a continuación [3].

- Se añade una red de revestimiento ad-hoc en la red celular de infraestructura fija.
- Los canales ad-hoc que se utilizan en esta red ayudan a la infraestructura fija a hacer un balance de carga.
- Los canales de la red celular de infraestructura fija se llaman canales RF.
 - Si los canales ad-hoc son de RF dentro de la banda celular, el sistema se llama in-band MACA.
 - Si los canales ad-hoc también son canales fuera de la banda celular, el sistema se llama out-of-band MACA.
- Los usuarios agentes de células vecinas *cold* son unidades intermediarias que conectan un usuario de la *hot cell* a la estación base de la *cold cell*.
- Para incrementar la reutilización de canales, los canales ad-hoc pueden tener una potencia menor y sólo alcanzar una distancia corta comparado con los canales RF.

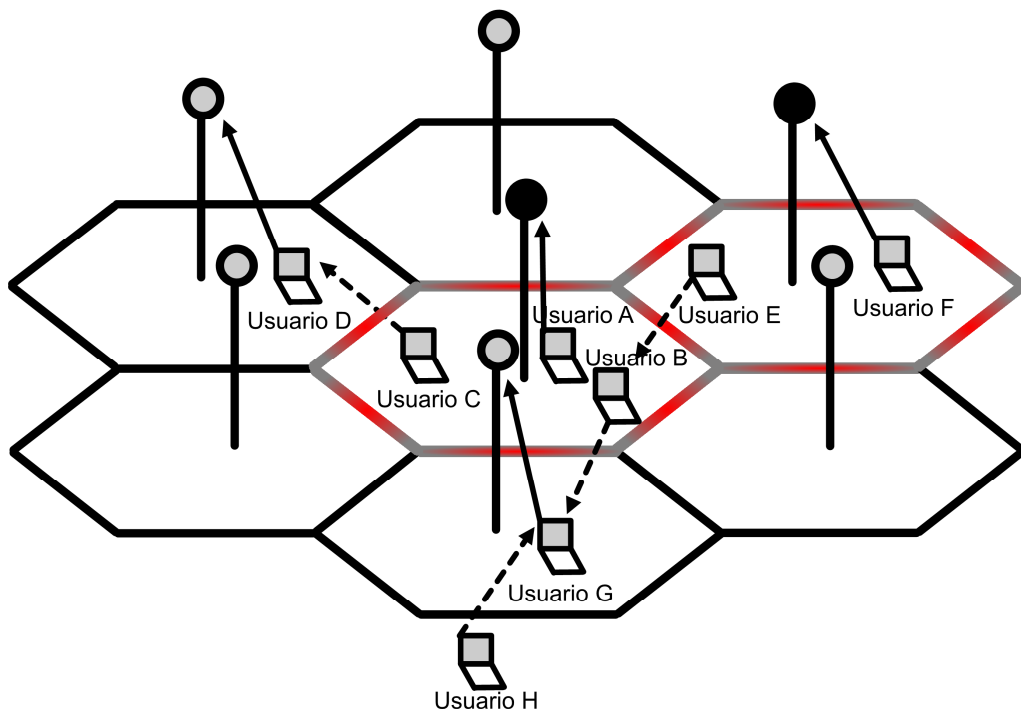


Fig. 13. Sistema celular compuesto por dos *hot cells*, varios usuarios y agentes para la técnica MACA.

La figura 13 muestra un sistema MACA con dos *hot cells*. Los canales dibujados con líneas continuas son los canales de RF, y los canales dibujados con líneas discontinuas son los canales ad-hoc. D y G son usuarios y B un agente que hace de

intermediario para conectar un usuario de una *hot cell* con la BS de una *cold cell*. MACA normalmente contiene dos saltos, pero también puede contener más de dos saltos (por ejemplo del usuario E al usuario B al usuario G a la BS del usuario G), en este caso se trata de multihop MACA. Un usuario fuera de cobertura de la red inalámbrica fija pero no demasiado lejos, como el usuario H, puede conectarse a la red por sí mismo tomando el usuario G como agente.

En lo que respecta a la operación del sistema tenemos en cuenta lo siguiente:

- El espectro para canales ad-hoc está dividido en dos partes: los canales de tráfico y los canales de señalización.
- En una red inalámbrica con MACA, un móvil al que le gustaría ser un agente, hará un broadcast de una señal libre a través del canal de señalización, la cual incluye potencia de señal, ID de agente y tráfico en la célula. Esta señal libre, sólo la pueden recibir los móviles dentro de la cobertura del canal ad-hoc.
- Normalmente, sólo los móviles en las *cold cells* con canales de RF libres pueden actuar como agentes, pero a veces cuando se necesita cruzar de célula con MACA, los móviles de las *hot cells* también pueden ser agentes.
- Un usuario recogerá estas señales libres para construir una tabla de agentes.
- Los agentes son elegidos por la estabilidad de la conexión con el usuario móvil.
- Cuando el usuario necesita usar MACA, elegirá un agente de su tabla de agentes, encontrará un canal ad-hoc apropiado (sin interferencia cocanal) y hará la conexión con este agente.
- Cuando el agente usado en MACA no es adecuado, porque la QoS medida con el BER (Bit Error Rate) en el canal ad-hoc se degrada, el usuario enviará una petición (*request*) a otro agente elegido de su tabla. A continuación se construye un nuevo link MACA y el otro se libera.
- La estación base (BS) tiene información de sus usuarios y agentes.
- Si un usuario que usa MACA se mueve fuera de la cobertura de los agentes o un recién llegado a la célula hot no alcanza ningún agente en la célula *cold*, la BS usará el channel switching, es decir, preguntará a uno de sus usuarios activos de la frontera de la célula para usar MACA y liberará el canal de RF.

2.3. Otras técnicas dinámicas de balance de carga.

Por último vamos a explicar dos técnicas dinámicas de balance de carga, LBSB y HCA, las cuales podemos considerar como un tipo especial de balance dinámico directo, pero requerirán un estudio distinto debido a sus características concretas. Cada una de ellas tendrá una serie de eventos y formas de proceder distintos al método directo.

2.3.1. Hybrid Channel Assignment (HCA).

A continuación exponemos las características principales del método HCA [7]:

- El número total de canales que tiene disponibles son la suma de un conjunto fijo y un conjunto dinámico.
- El conjunto fijo cuenta con un número de canales asignados a las células (como en la técnica FCA).
- El conjunto dinámico tiene un número de canales compartidos por todos los usuarios del sistema. A este conjunto de canales dinámicos se le llama *pool* de canales dinámicos.
- Si se inicia una llamada y hay canales nominales (fijos) libres, entonces se asigna un canal fijo a la llamada.
- Si una llamada requiere servicio de una célula y los canales nominales están ocupados, se asigna un canal dinámico a la llamada, siempre que existan canales dinámicos libres. Si la llamada no encuentra un canal disponible puesto que no quedan ni canales fijos ni dinámicos, entonces la llamada se bloquea.
- No se prestan canales entre las células, puesto que el grupo dinámico de canales hace la función de balance de carga.

2.3.2. Load Balancing with Selective Borrowing (LBSB).

LBSB [6] es un esquema de balance de carga que se basa en *borrowing* de canales. Propone migrar un número de canales de una *cold cell* a una *hot cell* mediante un algoritmo de préstamo de canales, antes de que el set de canales disponibles esté agotado. Las técnicas anteriores en las que se pedían prestados canales, necesariamente debían tener todos los canales de cada célula asignados a llamadas para proceder a asignar los canales prestados, pero en LBSB no es así. Utiliza una estrategia de asignación de canales, según tres tipos de usuarios: *new*, *departing*, *others*. Los canales locales y los prestables se asignan según la prioridad de los usuarios. Las células sobrecargadas obtienen el número necesario de canales, pero requiere una computación intensiva para la clasificación de usuario en célula.

Concretamente LBSB se trata de un sistema celular centralizado o distribuido aplicado para pocas células. En los esquemas centralizados, existe un servidor central donde se encuentra el algoritmo de balance de carga, el cual se actualiza según la información de estatus de los procesadores de la red. En los distribuidos, cada procesador es capaz de hacer funcionar el algoritmo de balance de carga. Sin embargo, puede haber una sobrecarga de mensajes de información de estatus entre los procesadores.

Las *hot cells* tienen el privilegio de pedir prestados canales y los usuarios *departing* tienen la máxima prioridad para pedir prestados los canales. Una *hot cell* pide prestados canales a una *cold cell* según una estrategia definida según varios parámetros como son: “*coldness*”, “*nearness*” y “bloqueo de canal de la *hot cell*”.

La BS periódicamente monitoriza la potencia de la señal recibida (RSS) por cada usuario. Cuando un usuario entra en una nueva célula, BS lo designa como usuario *new* y empieza un contador de τ unidades. El usuario continúa *new* si su estado no cambia durante ese periodo de tiempo. Si la RSS de un usuario es menor que un valor umbral, se considera que el usuario está en los alrededores de la célula. Si para este usuario de los alrededores de la célula comienza a disminuir su RSS, se empieza otro contador de α unidades. Si para este tiempo α su RSS disminuye, el usuario pasa a ser *departing*. Si no disminuye, el contador se resetea. Un usuario *others* es aquel que no es ni *new* ni *departing*. Un *departing* pasa a ser un usuario *others* si su RSS no disminuye durante 3 periodos de tiempo monitorizados por la BS. Un usuario *new* pasa a *others* cuando pasan τ unidades de tiempo.

En el momento que a un usuario se le asigna una llamada nueva, este usuario es *new*. Este usuario *new* puede pasar a ser *departing*. En un primer instante, el usuario debe tener un RSS menor que un umbral, para considerar que está en los alrededores o parte más externa de la célula. En este caso, si durante α unidades su RSS disminuye, el usuario pasa a ser *departing*. Si no, continua siendo *new* y vuelve a empezar con el contador de α unidades, y así sucesivamente. En el momento en que se cumplen τ unidades desde la asignación de la llamada, si el usuario *new* no ha pasado a ser *departing*, entonces pasa a ser *others*. Un usuario *others* también puede pasar a *departing* tras α unidades disminuyendo su RSS, una vez se consideró que estaba en los alrededores de la célula. Cuando un usuario es *departing* se puede asignar a la célula vecina si existen canales para él y la CIR recibida es mayor que en la primera célula (ver figura 14).

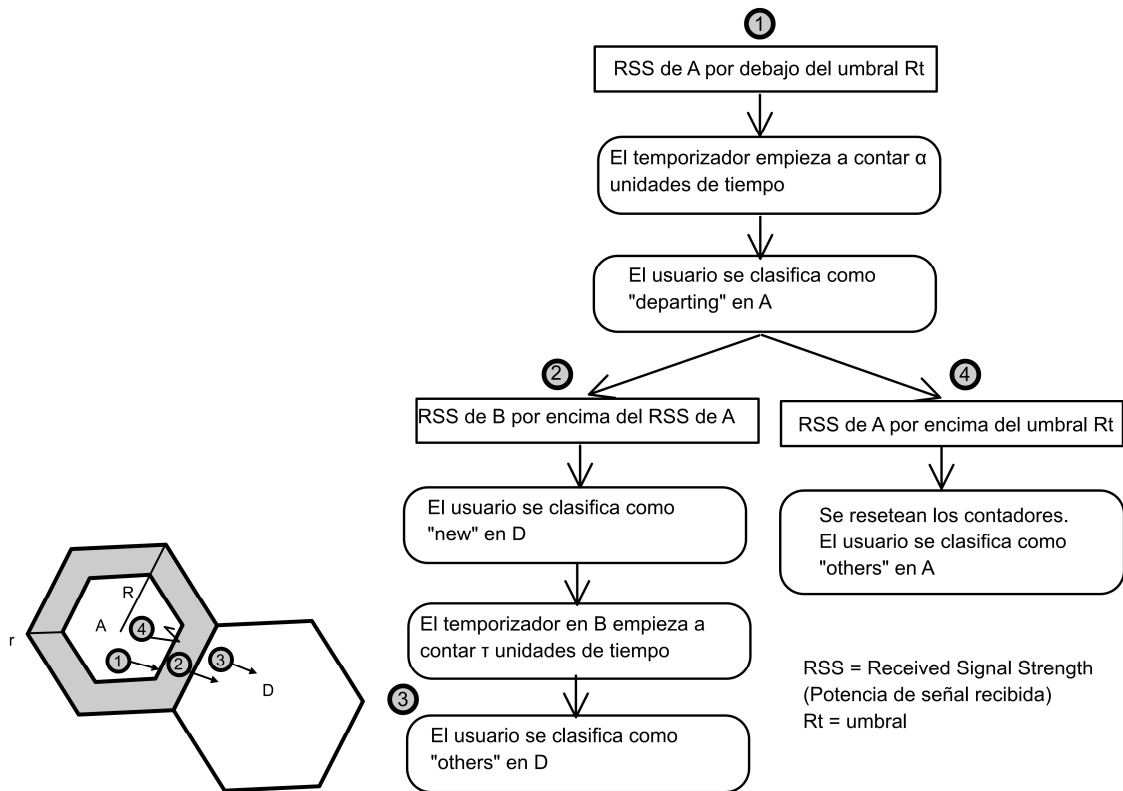


Fig. 14. Esquema de clasificación de usuarios en la técnica LBSB.

LBSB clasifica cuatro clases de demanda de canal según el tipo de usuario. La forma de proceder del algoritmo de asignación de canales cuando se solicita una llamada es distinta según la clase. Según cada clase se asignan canales locales o prestables, o se reasignan canales o bien se bloquean peticiones (ver figura 15).

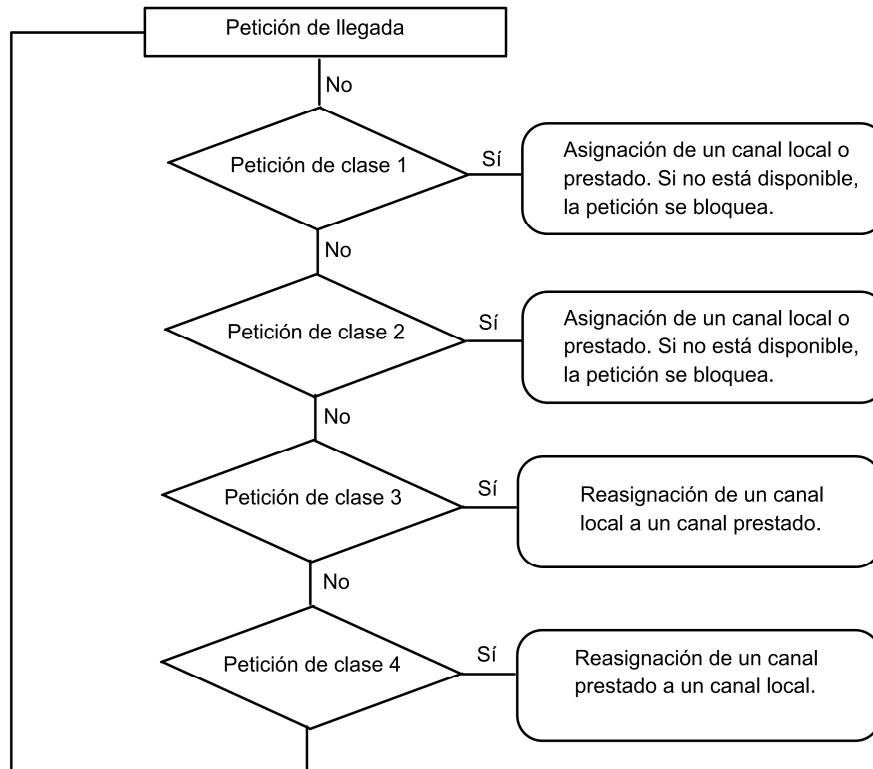


Fig. 15. Algoritmo de asignación de canales en la técnica LBSB.

3. Los métodos de balance de carga en un sistema de dos células

El estudio de un sistema de dos células se incluye para estudiar con precisión los mecanismos del método directo e indirecto en un escenario sencillo. Tiene una función didáctica y trata de descubrir de forma sencilla los fenómenos involucrados.

A continuación explicaremos el procedimiento que llevan a cabo los métodos directo e indirecto para el caso de un sistema compuesto por dos células. Asumimos que en este sistema no habrá interferencia cocanal, puesto que la reutilización de frecuencias se hace para distancias mayores que la distancia de reuso cocanal [6], la cual es mayor que la distancia entre dos células. En sistemas más extensos puede haber interferencia cocanal.

Primeramente no consideramos movilidad del usuario en la célula, es decir, suponemos que las llamadas que se asignan al usuario se inician y finalizan en el mismo punto geográfico. Este hecho simplifica el desarrollo del procedimiento a seguir.

Consideramos a continuación una serie de variables y definiciones para el estudio de ambos métodos.

- i_A e i_B es el número de llamadas en curso en la célula A y B respectivamente.
- C es el número de canales de cada célula. El número total de canales en las dos células será $2C$, independientemente de que haya canales que se usen para balance de carga. Mientras haya canales libres, no se utilizarán canales que participen en el balance de carga.
- B es el número de canales prestables que pueden ser usados para balance de carga, es decir, el número de canales que pueden ser prestados a otras células o bien el número de canales usados para enviar llamadas (forwarded calls) desde una célula a otra (es importante notar la diferencia entre canal prestable y canal prestado, aunque normalmente se hace referencia a canales prestados). C y B son funciones del ancho de banda disponible, el nivel de interferencia y el tipo de modulación.
- $C - B$ es el número de canales fijos de una célula.
- $C + B$ es el número máximo de canales usados en una célula.
- p es la probabilidad de que un nuevo usuario participe en un balance de carga y tiene distintos matices según la técnica que se esté utilizando.
 - p puede ser la probabilidad de que un nuevo usuario esté en el área de solape entre células (TTR de la parte fronteriza de las células), en el caso de DR.

- p puede ser la probabilidad de que el nuevo usuario esté dentro del área de potencia de transmisión reducida, es decir, la TTR alrededor de la BS, en el caso de CBWL.
- En HCB y MACA, se puede utilizar un canal para balance de carga en cualquier zona de la célula, por tanto p será igual a uno.
- j_A y j_B es el número de llamadas en curso (o usuarios activos) dentro de la TTR de la célula A y B. Este parámetro nos servirá para poder explicar el balance de carga indirecto.
- Una *hot cell* y una *cold cell* son aquellas células cuya probabilidad de bloqueo están por encima y por debajo de un valor objetivo, respectivamente.

3.1. El método directo de balance de carga en un sistema de dos células.

El método directo [1] es una técnica de balance de carga dinámico en la que solamente las nuevas peticiones de llamadas participan en el balance de carga.

A continuación presentamos en la figura 16, a modo de ejemplo, un sistema de dos células con la TTR en el área fronteriza de las células que usa el método directo, con $C=10$ canales para cada célula, $B=6$ canales que se pueden usar para balance de carga, $i_A=13$ llamadas en curso en la célula A e $i_B=5$ llamadas en curso en la célula B.

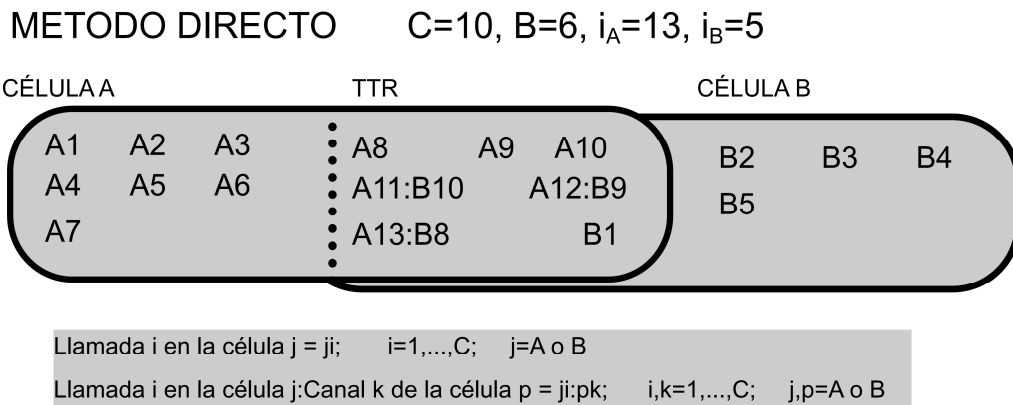


Fig. 16. Ejemplo de sistema de dos células que usa el método directo con TTR en la zona fronteriza.

Seguidamente evaluamos la dinámica del método directo cuando tenemos en cuenta los eventos llegada de una nueva llamada y terminación de una llamada en curso. Presentaremos algunas figuras que ayudan a comprender mejor cada situación.

- Si $0 \leq i_A < C$ y $i_A + i_B < 2C$ (ver figura 17):
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal con probabilidad 1. Se incrementa i_A en una unidad.
 - Si hay una llamada que termina, el canal queda liberado en esa célula. Se reduce i_A en una unidad.
 - No se utilizan canales prestables para balance de carga ya que el número de llamadas en curso es menor que C.

METODO DIRECTO $C=10, B=6, i_A=6$
 $i_A=8$

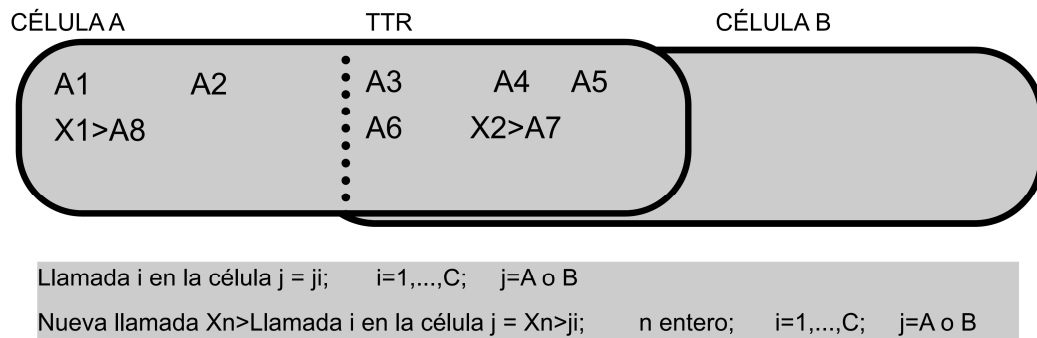


Fig. 17. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=6$ llamadas en curso en la célula A. Posteriormente, se inicia una llamada X_1 en TTR y otra X_2 fuera de TTR. En ambos casos se asigna una canal a cada llamada que se inicia puesto que quedan canales libres de la célula A que se pueden asignar, resultando $i_A=8$.

- Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal prestable que se usa para balance de carga con una probabilidad p . Se incrementa i_A en una unidad (ver figura 18).

METODO DIRECTO $C=10, B=6, i_A=13, i_B=5$
 $i_A=14, i_B=5$

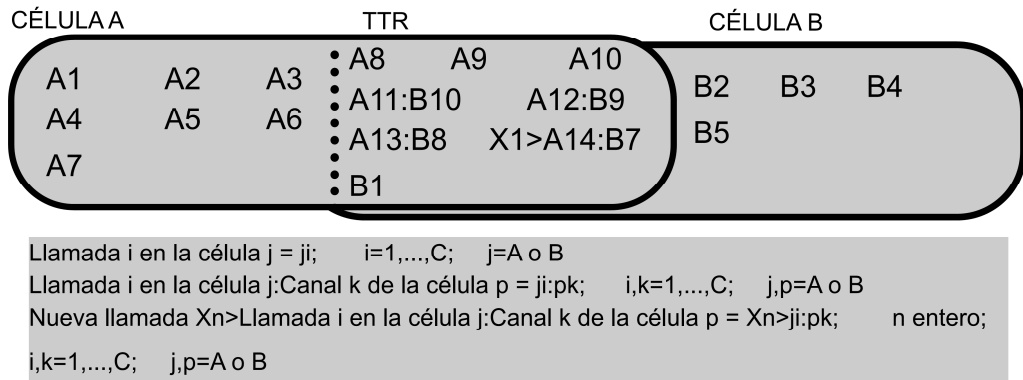


Fig. 18. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X_1 en TTR, resultando $i_A=14$, de las cuales 4 son servidas por la célula B.

- Si se inicia una llamada en la célula A, la llamada queda bloqueada con probabilidad $1 - p$. Aunque haya canales disponibles en la otra célula, el nuevo usuario no puede intervenir en el balance de carga (generalmente por la localización del usuario, ver figura 19).

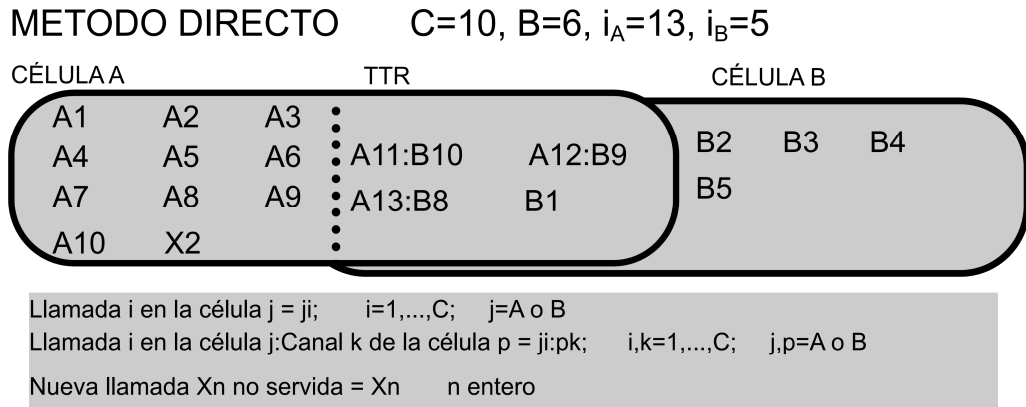


Fig. 19. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada X2 fuera de la TTR. La llamada no puede ser servida y queda bloqueada.

- Cuando una llamada que usa un canal de la célula A termina y existe algún usuario que usa un canal prestable de la célula B, se cambia un canal prestable por el canal liberado. Se reduce i_A en una unidad. De esta manera seguiremos teniendo los C canales asignados de la célula A, además de los canales usados para balance de carga. A este procedimiento se le llama *intra-handover* (ver figura 20).

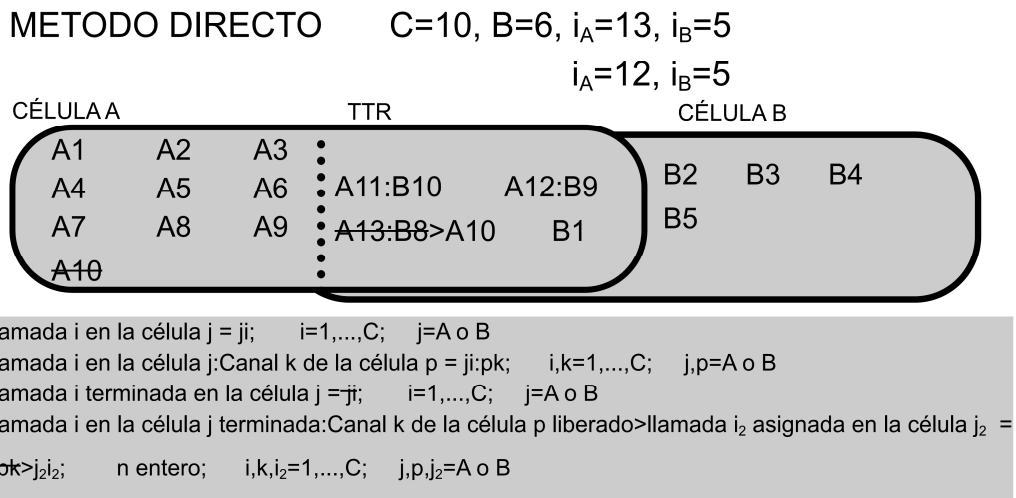
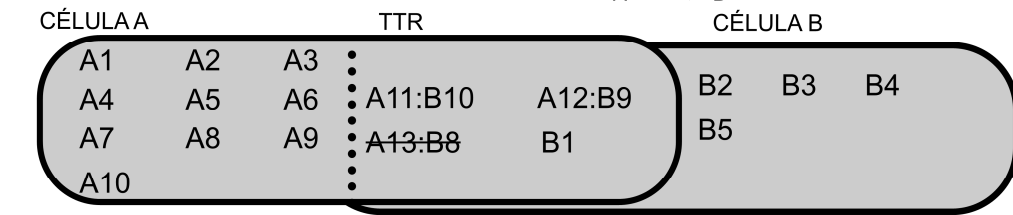


Fig. 20. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B (la llamada 11, 12 y 13). Posteriormente, se termina la llamada activa que usa el canal 10 de la célula A situada fuera de TTR y se intercambia dicho canal por el canal 8 prestado o servido por la célula B que usaba la llamada 13. De esta manera la llamada que usaba el el canal 8 de la célula B ahora usa el canal 10 propio de la célula A que ha sido liberado, resultando $i_A=12$, de las cuales únicamente 2 son servidas por la célula B.

- Si termina una llamada de la célula A que usa un canal usado en balance de carga, se liberará dicho canal prestable. Se reduce i_A en una unidad (ver figura 21).

METODO DIRECTO $C=10, B=6, i_A=13, i_B=5$

$i_A=12, i_B=5$



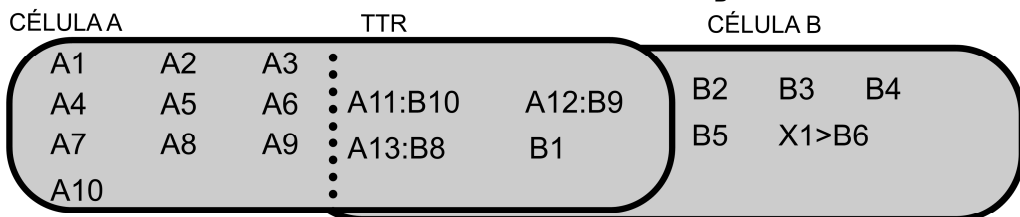
Llamada i en la célula $j = ji; \quad i=1, \dots, C; \quad j=A \text{ o } B$
 Llamada i en la célula j :Canal k de la célula $p = ji:pk; \quad i,k=1, \dots, C; \quad j,p=A \text{ o } B$
 Llamada i en la célula j terminada:Canal k de la célula p liberado = $ji:pk; \quad n$ entero; $i,k=1, \dots, C; \quad j,p=A \text{ o } B$

Fig. 21. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A, de las cuales 3 son servidas por la célula B. Posteriormente, termina la llamada 13 de la célula A que utiliza el canal 8 prestado por B, resultando $i_A=12$.

- Si se inicia una llamada en la célula B se asignará un canal libre con probabilidad 1, no un canal usado para balance de carga puesto que a la célula B todavía le quedan libres. Se incrementa i_B en una unidad (ver figura 22).

METODO DIRECTO $C=10, B=6, i_A=13, i_B=5$

$i_B=6$



Llamada i en la célula $j = ji; \quad i=1, \dots, C; \quad j=A \text{ o } B$
 Llamada i en la célula j :Canal k de la célula $p = ji:pk; \quad i,k=1, \dots, C; \quad j,p=A \text{ o } B$
 Nueva llamada $Xn >$ Llamada i en la célula $j = Xn>ji; \quad n$ entero; $i=1, \dots, C; \quad j=A \text{ o } B$

Fig. 22. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A e $i_B=5$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada $X1$ en la célula B fuera de la TTR, resultando $i_B=6$.

- Si $i_A = C + B$ y $i_A + i_B < 2C$:
 - Si se inicia una llamada en la célula A, la llamada será bloqueada con probabilidad 1, ya que no es posible encontrar un canal libre para esta llamada (ver figura 23).

METODO DIRECTO $C=10, B=6, i_A=16, i_B=3$

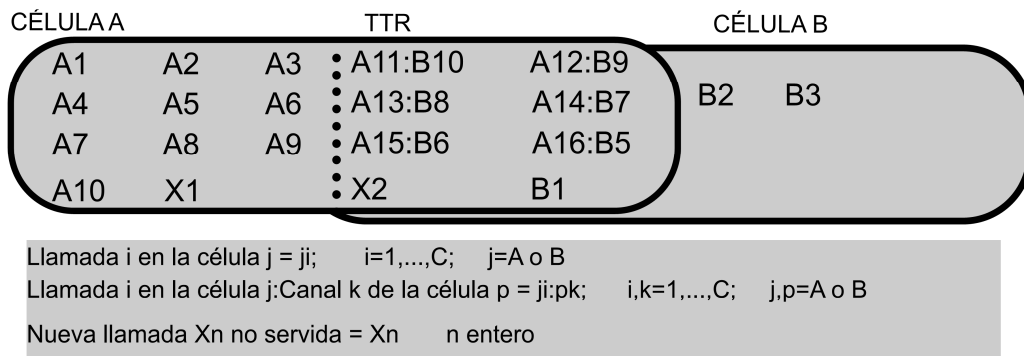


Fig. 23. Ejemplo de sistema de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=16$ llamadas en curso en la célula A e $i_B=3$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada X1 en la célula A fuera de la TTR y otra llamada X2 en la TTR. Las dos llamadas quedan bloqueadas. X1 y X2 se bloquean porque el máximo número de canales que se puede asignar a una célula es $C+B$.

- Se puede terminar una llamada en la célula A o B. El comportamiento cuando termina una llamada en una célula que utiliza canales para balance de carga es el mismo que el caso anterior. Si no los utiliza, se libera un canal propio de la célula. Se reduce i_A o i_B en una unidad.
- Si ocurre una llamada en la célula vecina B podrá asignarle un canal de la célula B con probabilidad 1. Se incrementa i_B en una unidad.
- Si $i_A + i_B = 2C$:
 - El sistema está congestionado. No podemos utilizar ni canales propios de la célula ni canales para balance de carga.
 - Si se inicia una llamada en la célula A o B, la llamada será bloqueada con probabilidad 1, debido a que no es posible encontrar un canal libre para una llamada iniciada en cualquiera de las células (ver figura 24).

METODO DIRECTO $C=10, B=6, i_A=16, i_B=4$

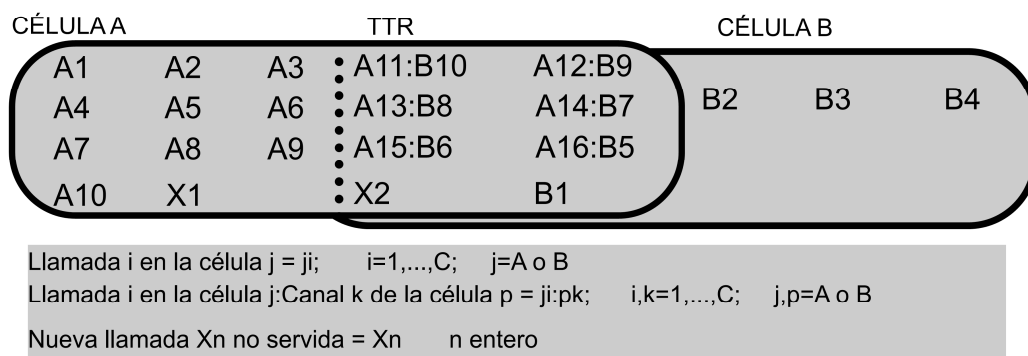


Fig. 24. Ejemplo de sistema congestionado de dos células que usa el método directo, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=16$ llamadas en curso en la célula A e $i_B=4$ llamadas en curso en la célula B. Posteriormente, se inicia una llamada X1 en la célula A y otra llamada X2 en la célula B. Las dos llamadas quedan bloqueadas.

- Sólo se puede terminar una llamada en la célula A o B. Se reduce i_A o i_B en una unidad.
- En una célula congestionada A (con $i_A = C + B$), existen B llamadas que usan un canal prestable de la célula B.

A continuación estudiaremos las técnicas que emplean el método directo para un sistema de dos células, que como hemos comentado son Hybrid Channel Borrowing (HCB), Channel Borrowing Without Locking (CBWL) y Directed Retry (DR).

3.1.1. Hybrid Channel Borrowing (HCB).

Seguidamente exponemos las características que distingue a HCB de los demás métodos directos para un sistema de dos células. Los eventos y parámetros son exactamente igual que los ya explicados en el apartado anterior.

Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:

- Si se inicia una llamada en la célula A, se asigna la llamada a un canal que nos presta la célula B con probabilidad igual a 1.
- El canal prestado por la célula B es servido por la BS de la célula A.
- La llamada nunca se bloquea, puesto que $p=1$.

3.1.2. Channel Borrowing Without Locking without channel Rearrangement (CBWL/NR).

Las características que distinguen a CBWL/NR de los demás métodos directos para un sistema de dos células son las siguientes:

Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:

- Si se inicia una llamada en la célula A, se asigna la llamada a un canal que nos presta la célula B vecina con probabilidad $p < 1$.
- El canal prestado por la célula B es servido por la BS de la célula A.
- Si se inicia una llamada en la célula A, la llamada queda bloqueada con probabilidad $1 - p$.

3.1.3. Directed Retry (DR).

A continuación exponemos las características que distingue a DR de los demás métodos directos para un sistema bicelular.

Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:

- Si se inicia una llamada en la célula A, se envía la llamada a la célula vecina (forwarded calls), con una probabilidad $p < 1$.
- El canal prestado por la célula B es servido por la BS de la célula B.
- Si se inicia una llamada en la célula A, la llamada queda bloqueada con probabilidad $1 - p$.

3.2. El método indirecto de balance de carga en un sistema de dos células.

El método indirecto en un sistema de dos células, presenta semejanzas respecto al método directo. Los parámetros utilizados para explicarlo son los mismos añadiendo j_A y j_B , que son el número de llamadas en curso (o usuarios activos) dentro de la TTR de la célula A y B.

A continuación presentamos en la figura 25, a modo de ejemplo, una situación de dos células con una TTR en la parte fronteriza de las células para el método indirecto, con $C=10$ canales para cada célula, $B=6$ canales que se pueden usar para balance de carga, $i_A=13$ llamadas en curso en la célula A, $i_B=5$ llamadas en curso en la célula B, $j_A=6$ llamadas en curso en la zona TTR de la célula A y $j_B=1$ llamadas en curso en la TTR de la célula B.

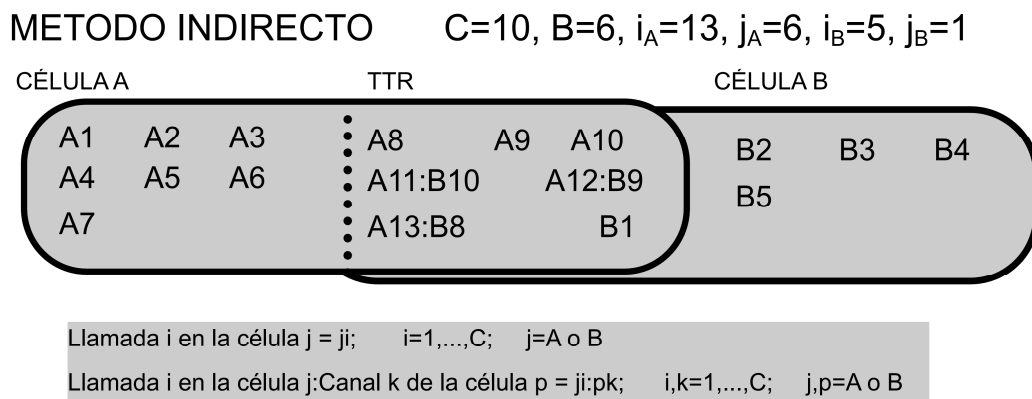


Fig. 25. Ejemplo de sistema de dos células que usa el método indirecto con TTR en la zona fronteriza.

Seguidamente evaluamos la dinámica del método indirecto cuando tenemos en cuenta los eventos llegada de una nueva llamada y terminación de una llamada en curso. Al igual que en el método directo, presentaremos algunas figuras que ayudan a comprender mejor cada situación.

- Si $0 < i_A < C$ y $i_A + i_B < 2C$:
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal con probabilidad 1. Se incrementa i_A en una unidad.
 - Si hay una llamada que termina, el canal queda liberado en esa célula. Se reduce i_A en una unidad.
 - No se utilizan canales prestables ya que el número de llamadas en curso es menor que C.
 - Una nueva llamada puede ser utilizada para el balance de carga del método indirecto con una probabilidad p . En este caso se incrementa j_A en una unidad (ver figura 26).

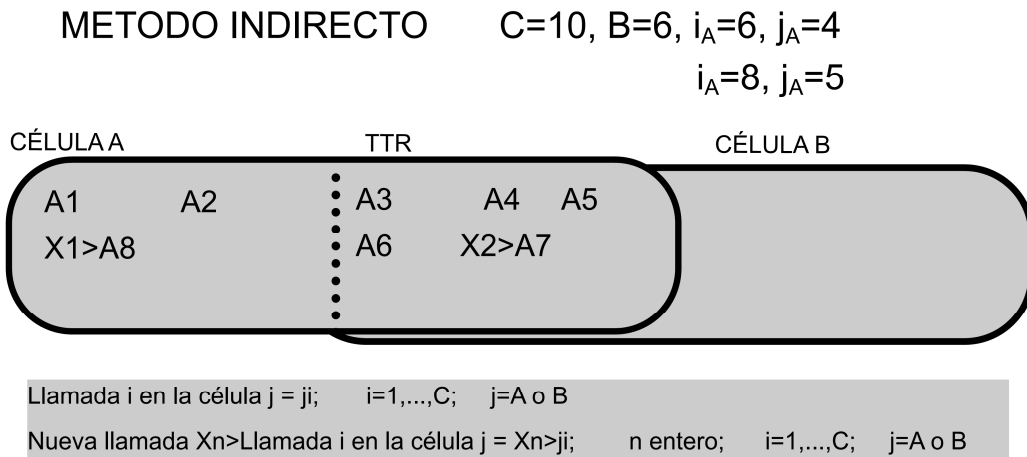


Fig. 26. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=6$ llamadas en curso en la célula A y $j_A=4$ llamadas en curso en la TTR. Posteriormente, se inicia una llamada $X2$ en TTR y otra $X1$ fuera de TTR, resultando $i_A=8$ y $j_A=5$.

- Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal prestable que se usa para balance de carga con una probabilidad p . Se incrementa i_A y j_A en una unidad (ver figura 27).

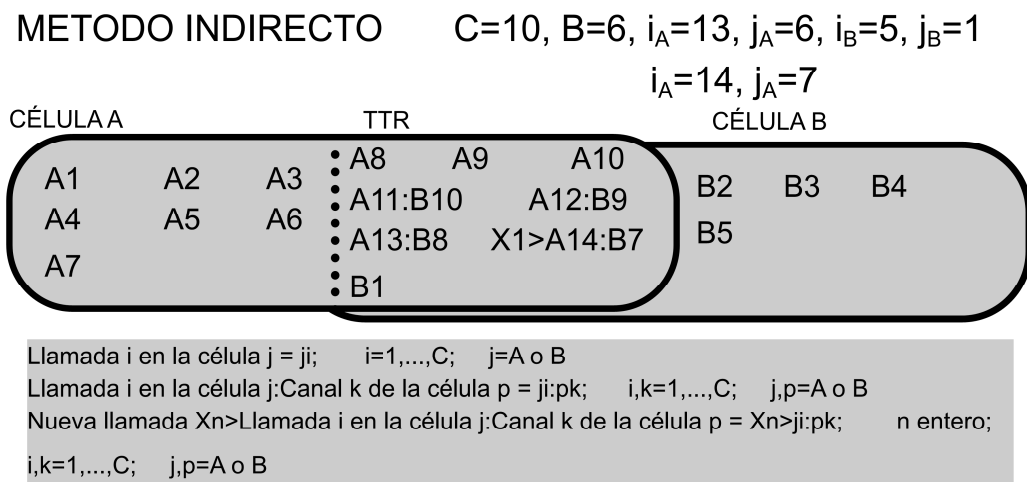


Fig. 27. Ejemplo de sistema de dos células que usa el método indirecto, con $C=10$ canales y $B=6$ canales. Inicialmente hay $i_A=13$ llamadas en curso en la célula A y $j_A=6$ llamadas en la TTR de las cuales 3 son servidas por la célula B. Posteriormente, se inicia una llamada $X1$ en TTR, resultando $i_A=14$ y $j_A=7$, de las cuales 4 son servidas por la célula B.

- Si se inicia una llamada en la célula A, la llamada queda fuera de la TTR con probabilidad $1 - p$. Si la llamada está fuera de este área, y $j_A > i_A - C$; entonces se puede cambiar el canal de la llamada activa a un canal prestable, para liberar el canal que usaba y entregárselo a la nueva llamada fuera de la zona de TTR que pedía servicio (ver figura 28). Se incrementa i_A en una unidad. En caso de que $j_A = i_A - C$ y la nueva llamada esté fuera de la TTR, la llamada quedará bloqueada (ver figura 29).

- Si $i_A = C + B$ y $i_A + i_B < 2C$:
 - Si se inicia una llamada en la célula A, la llamada será bloqueada con probabilidad 1, ya que no es posible encontrar un canal libre para esta llamada.
 - Se puede terminar una llamada en la célula A o B. El comportamiento cuando termina una llamada en una célula que utiliza canales para balance de carga es el mismo que el caso anterior. Si no los utiliza, se libera un canal propio de la célula. Se reduce i_A o i_B en una unidad.
 - Si ocurre una llamada en la célula vecina, B podrá asignarle un canal de la célula B. Se incrementa i_B en una unidad.
- Si $i_A + i_B = 2C$:
 - El sistema está congestionado. No podemos utilizar ni canales propios de la célula ni canales para balance de carga.
 - Si se inicia una llamada en la célula A o B, la llamada será bloqueada con probabilidad 1, debido a que no es posible encontrar un canal libre para una llamada iniciada en cualquiera de las células.
 - Sólo se puede terminar una llamada en la célula A o B. Se reduce i_A o i_B en una unidad.
 - En una célula congestionada 1 (con $i_A = C + B$), existen B llamadas que, o bien usan un canal prestado de la célula B (como en CBWL/CR) o bien están siendo servidas por la BS de la célula vecina B (MACA).

A continuación estudiaremos las técnicas que emplean este método indirecto para un sistema bicelular, las cuales son Channel Borrowing Without Locking with Channel Rearrangement (CBWL/CR) y Mobile-Assisted Call Admission algorithm (MACA).

3.2.1. Channel Borrowing Without Locking with Channel Rearrangement (CBWL/CR).

Seguidamente exponemos las características que distingue a CBWL/CR de los demás métodos indirectos para un sistema de dos células. Los demás eventos y parámetros son exactamente igual que los ya explicados en apartados anteriores.

Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:

- Si se inicia una llamada en la célula A, se asigna la llamada a un canal que nos presta la célula B vecina con una probabilidad $p < 1$.
- El canal prestado por la célula B es servido por la BS de la célula A.

- Si se inicia una llamada en la célula A, la llamada queda fuera de la TTR con probabilidad $1 - p$.

3.2.2. Mobile-Assisted Call Admission algorithm (MACA).

A continuación exponemos las características que le distingue de los otros métodos indirectos para dos células.

Si $C + B > i_A \geq C$, e $i_A + i_B < 2C$:

- Si se inicia una llamada en la célula A, se envía la llamada a la célula vecina (forwarded calls), con una probabilidad igual a 1.
- El canal prestado por la célula B es servido por la BS de la célula B.
- La llamada nunca se bloquea, puesto que $p=1$.

3.3. Otras técnicas dinámicas de balance de carga en un sistema de dos células.

3.3.1. Hybrid Channel Assignment (HCA).

El método HCA para un sistema de dos células considera algunos parámetros distintos a los explicados para las técnicas del método directo e indirecto. Llamamos N al número máximo de canales nominales de cada célula y D al número máximo de canales dinámicos. A modo de ejemplo exponemos en la figura 30, una situación de dos células para el método HCA, con $N=10$ canales fijos para cada célula, $D=20$ canales dinámicos ubicados en el *pool*, $i_A=9$ llamadas en curso en la célula A e $i_B=9$ llamadas en curso en la célula B.

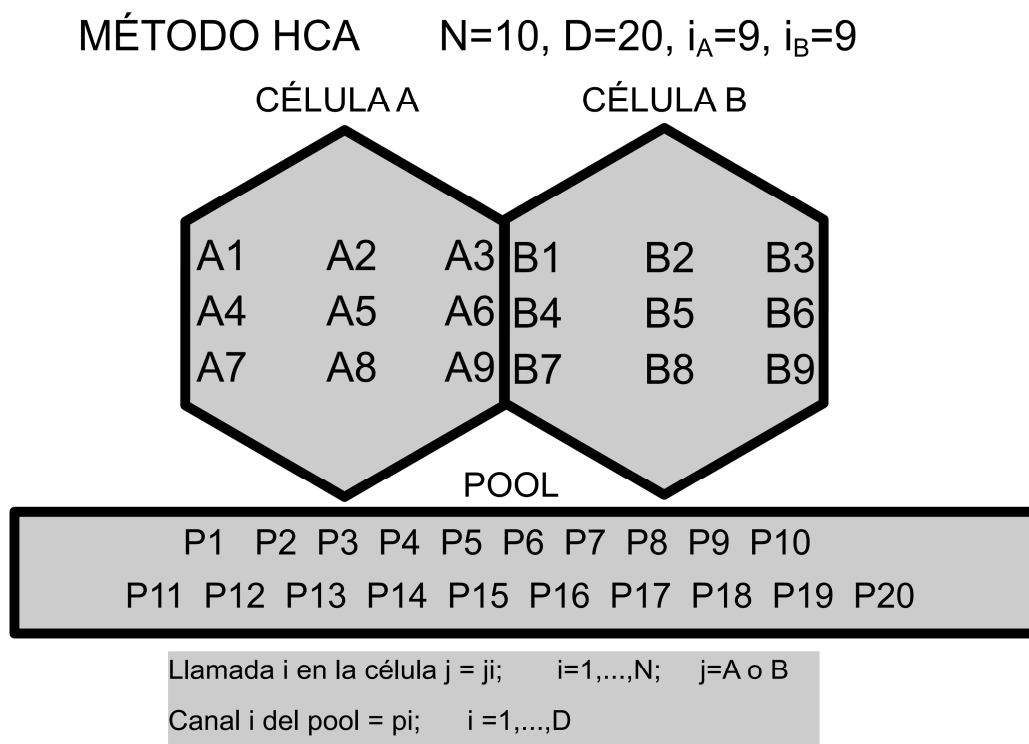


Fig. 30. Sistema compuesto por dos células que usa el método HCA.

A continuación evaluamos la dinámica del método HCA cuando tenemos en cuenta los eventos llegada de una nueva llamada y terminación de una llamada en curso. Mostraremos figuras ejemplo de cada situación que consideramos.

- Si $0 < i_A < N$ y $0 < i_B < N + D$ (ver figura 31):
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal fijo con probabilidad 1. i_A se incrementa en una unidad.
 - Si se inicia una llamada en la célula B, i_B se incrementa en una unidad.
 - Si $i_B < N$ se asigna la llamada a un canal fijo con probabilidad 1.
 - Si $N + D > i_B > N$ se asigna la llamada a un canal dinámico con probabilidad 1.
 - Si hay una llamada que termina, el canal fijo o dinámico queda liberado en esa célula. i_A o i_B se reduce en una unidad.

MÉTODO HCA $N=10, D=20, i_A=10, i_B=8$

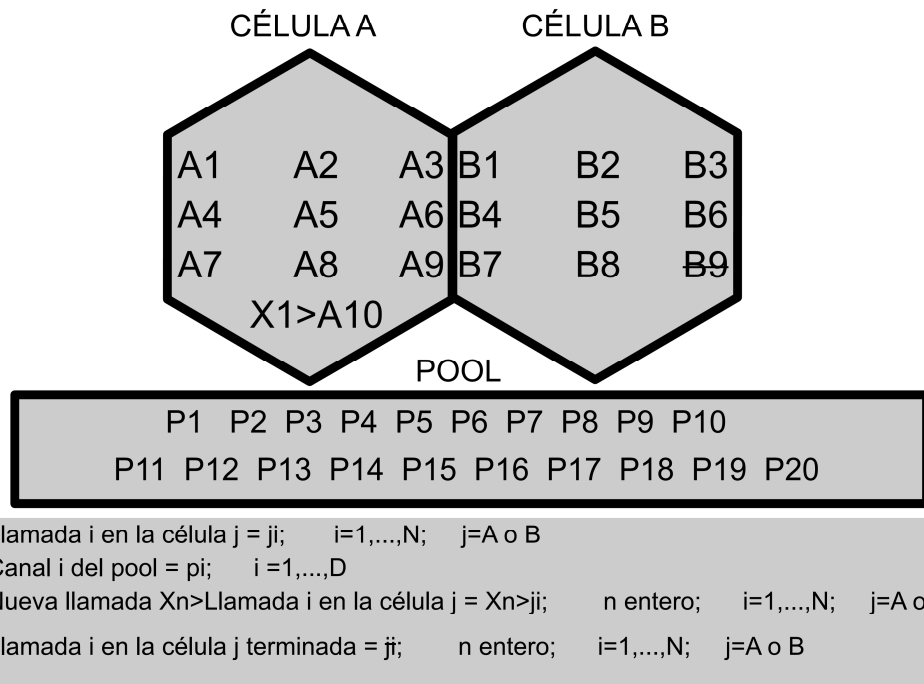
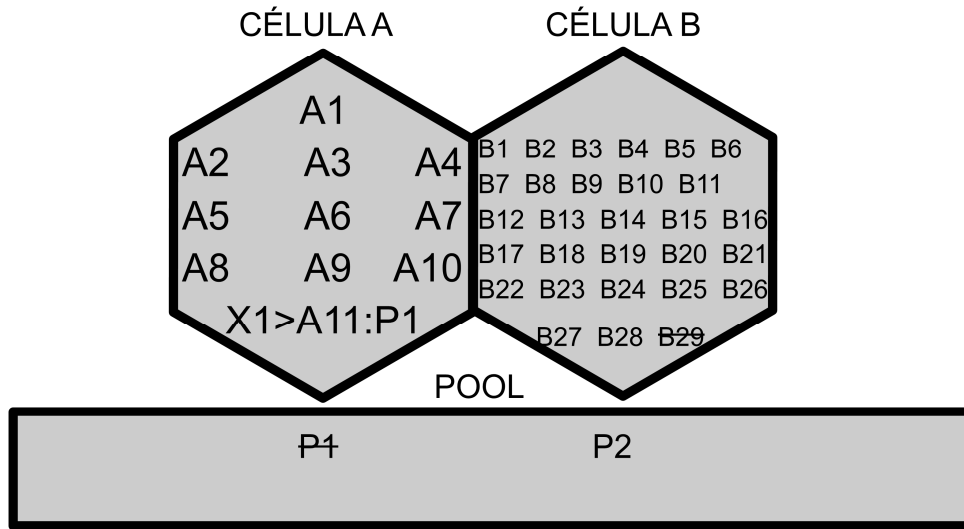


Fig. 31. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$. Inicialmente $i_A=9$ e $i_B=9$. Posteriormente, se inicia una llamada en la célula A y termina una llamada en la célula B. Los valores ahora son $i_A=10$ e $i_B=8$. En este caso los canales del *pool* no se han asignado.

- Si $N \leq i_A < N + D$ y $0 < i_B < 2N + D - i_A$ (ver figura 32):
 - Si se inicia una llamada en la célula A, se asigna la llamada a un canal dinámico.
 - La llamada nunca se bloquea.
 - Cuando una llamada que usa un canal fijo termina en cualquiera de las dos células y existe algún usuario que usa un canal dinámico, se cambiará un canal dinámico por el canal fijo liberado.
 - Si termina una llamada que usa un canal dinámico, se liberará dicho canal.
 - Si se inicia una llamada en la célula B
 - Si $i_B < N$, se asigna un canal fijo.
 - Si $D + N > i_B > N$, se asigna un canal dinámico.

MÉTODO HCA $N=10, D=20, i_A=10, i_B=29$
 $i_A=11, i_B=28$



Llamada i en la célula $j = ji$; $i=1, \dots, N$; $j=A$ o B
 Llamada i en la célula j terminada = ji ; n entero; $i=1, \dots, N$; $j=A$ o B
 Canal x liberado en el pool = px ;
 Nueva llamada $Xn >$ Llamada i en la célula j : canal k prestado por el pool = $Xn > ji: pk$;
 n entero; $i=1, \dots, N$; $k=1, \dots, D$; $j=A$ o B
 Canal k prestado por el pool = pk ; $k=1, \dots, D$

Fig. 32. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$. Inicialmente $i_A=10$ e $i_B=29$. Posteriormente, se inicia una llamada en la célula A y termina una llamada en la célula B. Los valores ahora son $i_A=11$ e $i_B=28$. La llamada X1 usa el canal que quedaba en el *pool*, y la llamada 29 libera el canal que utilizaba (P2) y lo asigna al *pool*.

- Si $i_A + i_B = 2N + D$ (ver figura 33):
 - El sistema está congestionado, todos los canales está siendo utilizados.
 - Si se inicia una llamada en la célula A o B, la llamada será bloqueada con probabilidad 1, ya que no es posible encontrar un canal libre para una llamada iniciada en cualquiera de las células.
 - Sólo se puede terminar una llamada en la célula A o B.
 - Cuando una llamada que usa un canal fijo termina en cualquiera de las células congestionadas, se cambiará un canal dinámico por el canal fijo liberado si alguna célula tiene un canal dinámico asignado, es decir, $i_A \geq N$ y/o $i_B \geq N$. Si alguna de ellas tiene sólo canales fijos asignados y ningún canal dinámico establecido, entonces simplemente se liberará el canal fijo.
 - Si termina una llamada que usa un canal dinámico, se liberará dicho canal.

MÉTODO HCA $N=10, D=20, i_A=10, i_B=30$
 $i_A=11, i_B=28$

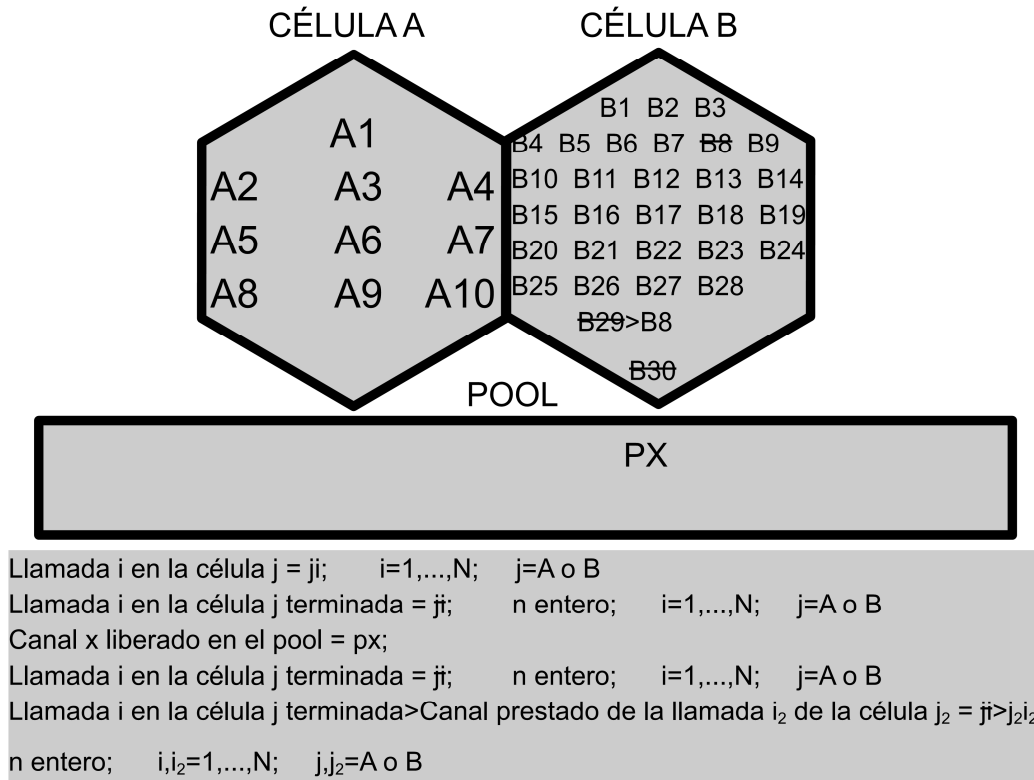


Fig. 33. Sistema compuesto por dos células que usa el método HCA con $N=10$ y $D=20$.

Inicialmente $i_A=10$ e $i_B=30$. Posteriormente, terminan dos llamadas en la célula B, la llamada 30 y la llamada 8. Los valores ahora son $i_A=10$ e $i_B=28$. La llamada 30 libera el canal dinámico que utilizaba y lo asigna al *pool*, la llamada 29 que usa un canal dinámico, cambia dicho canal por el canal 8 fijo liberado.

3.3.2. Load Balancing with Selective Borrowing (LBSB).

Todo el estudio de movilidad en LBSB no va a ser tema de estudio en esta primera parte, pues consideramos que las llamadas se inician y finalizan en el mismo punto geográfico y LBSB basa toda su estrategia en la movilidad de los usuarios entre células. No tenemos en cuenta la clase de demanda de canal, el tipo de los usuarios, ni por tanto si tienen más o menos prioridad. Debido a eso, hasta este punto LBSB se va a comportar como el esquema dinámico Hybrid Channel Borrowing (HCB), que ya hemos explicado anteriormente.

3.4 Descripción del modelo analítico en un sistema de dos células.

El método analítico se basa en el estudio de las técnicas de balance de carga dinámico mediante métodos matemáticos que se ajustan a su funcionamiento en diversos sistemas. Primero estudiaremos un sistema compuesto por dos células en el que no existe movilidad de las llamadas.

Asumimos llegadas de Poisson y tiempos de servicio exponencialmente distribuidos. Con ello podemos describir el número de llamadas en curso mediante un proceso de Markov [10] [11]. Los procesos de Markov obtenidos serán bidimensionales o de cuatro dimensiones, según trabajemos con el método directo o indirecto, respectivamente.

Los procesos de Markov que representan el sistema permiten obtener las probabilidades de estado y la probabilidad de pérdidas. Los procesos de más de una dimensión se pueden resolver mediante la metodología de los procesos reversibles. Para ello debemos justificar que el proceso es reversible, es decir, debemos comprobar que se cumplen las ecuaciones de balance de estados (ver X1) y aplicar la metodología de resolución.

$$\pi_i q_{ij} = \pi_j q_{ji} \quad (X1)$$

π_i y π_j son las distribuciones estacionarias de los sistemas *forward* y reversible, y q_{ij} y q_{ji} las tasas de transición de los procesos *forward* y reversible.

A partir de las ecuaciones de balance de estados podemos enunciar el criterio de Kolmogorov: “Un proceso de Markov es reversible si y solo si para cada ciclo de estados se verifica X2”

$$q(s_1, s_2)q(s_2, s_3) \dots q(s_m, s_1) = q(s_1, s_m)q(s_m, s_{m-1}) \dots q(s_2, s_1) \quad (X2)$$

La resolución de los procesos de más de una dimensión queda simplificada pues existe una propiedad de los procesos reversibles que asegura que si dos procesos reversibles X_1 y X_2 son independientes, el proceso $Y=(X_1, X_2)$ es reversible y se cumple que $P [Y] = P [X_1, X_2] = P [X_1] P [X_2]$.

Para los procesos bidimensionales X_1 y X_2 son las variables número de llamadas en cada célula y para los sistemas de cuatro dimensiones tenemos cuatro variables independientes, el número de llamadas en curso en cada célula y el número de llamadas en curso que pueden ser usadas en el balance de carga en cada célula.

3.4.1. Descripción del modelo analítico en un sistema de dos células para el método directo.

Para el método directo, los estados del proceso de Markov se definen como $S(i_A, i_B)$, donde i_A e i_B son el número de llamadas en curso en las células A y B.

El tamaño y forma de los procesos de Markov para el método directo (ver figura 34) en un sistema de dos células depende de varios parámetros, como son el número de canales de cada célula, C , el número de canales prestables, B , la probabilidad de que un nuevo usuario participe en un balance de carga, p , las tasas de llegadas de ambas células, λ_A y λ_B , y las tasas de servicio de las células, μ_A y μ_B .

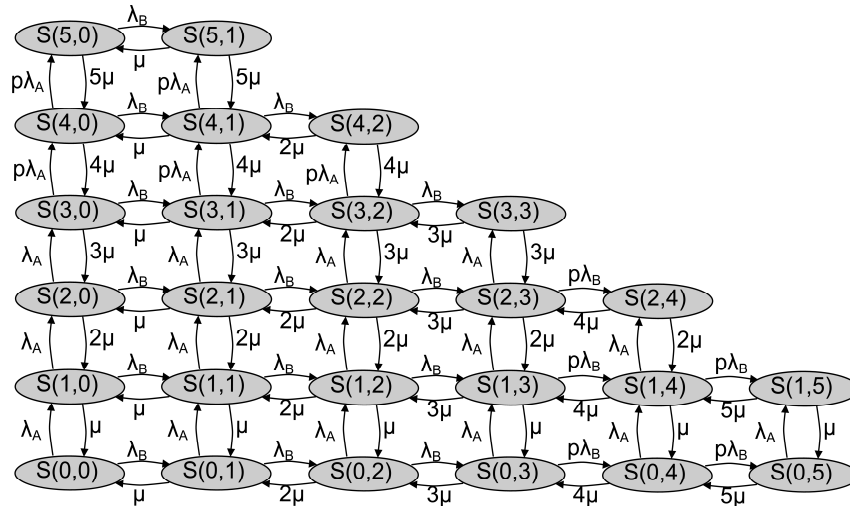


Fig.34. Diagrama de estados en un sistema ejemplo de dos células con $C = 3$ y $B = 2$ en el balance de carga indirecto.

Mediante los procesos de Markov podemos calcular las probabilidades de estado y la probabilidad de pérdidas de los sistemas de forma sencilla y mecánica.

La forma de obtener dichas probabilidades se hace mediante un método recursivo, apoyándonos en el hecho de que los procesos son reversibles, aunque también podríamos resolverlo con las ecuaciones de balance global.

Una vez hemos comprobado que se trata de un proceso reversible, realizamos un programa en Matlab para el método directo (y otro programa para el método indirecto) que resuelva las ecuaciones de balance de estados. Con la reversibilidad evitamos carga computacional.

Una vez hemos obtenido las probabilidades de estado, determinamos cuando una nueva petición de llamada será bloqueada en nuestro sistema:

- Una nueva llamada en la célula A será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + B$, están siendo utilizados.
 - Todos los canales de las células A y B, $2C$, están siendo utilizados.
- Una nueva llamada será bloqueada con probabilidad $(1 - p)$ si

- Todos los canales de la célula A se utilizan y, aunque hay canales disponibles para ser prestados, el nuevo usuario no puede participar en el balance de carga, debido a su localización.

El siguiente paso consiste en determinar cuales son los eventos de bloqueo de llamada en la célula A y a qué estados corresponden:

- $S(C+B, i_B)$ donde $0 \leq i_B \leq C - B$
- $S(i_A, i_B)$, donde $i_A + i_B = 2C$, $i_A \neq C + B$
- $S(i_A, i_B)$, donde $C \leq i_A < C + B$, $0 \leq i_B < 2C - i_A$.

Por último determinamos la probabilidad de bloqueo de llamada en la célula A, sumando los estados de bloqueo anteriormente descritos (ver X3):

$$\begin{aligned}
 P_{B_D} = & \sum_{i_B=0}^{C-B} S(C+B, i_B) + \sum_{i_B=C+1}^{C+B} S(2C-i_B, i_B) \\
 & + \sum_{i_A=C}^{C+B-1} S(i_A, 2C-i_A) + \sum_{i_A=C}^{C+B-1} \sum_{i_B=0}^{2C-i_A-1} S(i_A, i_B)(1-p)
 \end{aligned} \tag{X3}$$

3.4.2. Descripción del modelo analítico en un sistema de dos células para el método indirecto.

Para el método indirecto, los estados del proceso de Markov se definen como $S(i_A, i_B, j_A, j_B)$, donde i_A e i_B son el número de llamadas en curso en las células A y B, y j_A, j_B , el número de llamadas en curso en la zona de load balancing de las células A y B.

El tamaño y forma de los procesos de Markov (ver figura 35) para el método indirecto en un sistema de dos células también depende de los mismos parámetros que el método directo, como son C, B, p, λ_A y λ_B , y μ_A y μ_B .

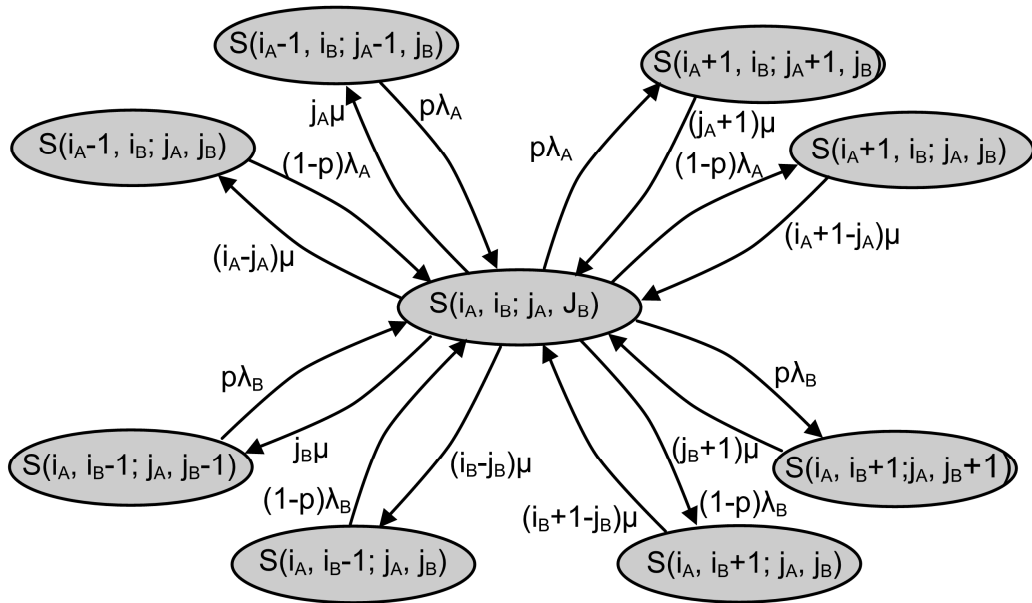


Fig. 35. Diagrama de estados en un sistema de dos células en el balance de carga indirecto.

Mediante los procesos de Markov podemos calcular las probabilidades de estado y la probabilidad de pérdidas de los sistemas.

La forma de obtener dichas probabilidades se hace también mediante un método recursivo, apoyándonos en la propiedad de reversibilidad de los procesos. Se puede resolver mediante las ecuaciones de balance global pero es un trabajo laborioso y computacionalmente es poco eficiente.

Una vez hemos obtenido las probabilidades de estado, determinamos cuando una nueva petición de llamada será bloqueada en nuestro sistema:

- Una nueva llamada en la célula A será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + B$, están siendo utilizados
 - Todos los canales de las células A y B, $2C$, están siendo utilizados.
- Una nueva llamada será bloqueada con probabilidad $(1 - p)$ si
 - Todos los canales de la célula A se utilizan y, aunque hay canales disponibles para ser prestados, no hay llamadas activas para emplear en load balancing.

El siguiente paso consiste en determinar cuales son los eventos de bloqueo de llamada en la célula A y a qué estados corresponden:

- $S(C+B, i_B; j_A, j_B)$ donde $0 \leq i_B \leq C - B$, $i_A - C \leq j_A \leq i_A$, $0 \leq j_B \leq i_B$

- $S(i_A, i_B; j_A, j_B)$, donde $i_A + i_B = 2C$, $C \leq i_A < C + B$, $i_B = 2C - i_A$, $i_A - C \leq j_A \leq i_A$, $0 \leq j_B \leq i_B$ o bien $C < i_B \leq C + B$, $i_A = 2C - i_B$, $i_B - C \leq j_B \leq i_B$, $0 \leq j_A \leq i_A$
- $S(i_A, i_B; i_A - C, j_B)$, donde $C \leq i_A < C + B$, $0 \leq i_B < 2C - i_A$, $0 \leq j_B \leq i_B$

Por último determinamos la probabilidad de bloqueo de llamada en la célula A, sumando los estados de bloqueo anteriormente descritos (ver X4):

$$\begin{aligned}
 P_{B_{ID}} = & \sum_{i_B=0}^{C-B} \sum_{j_B=0}^{i_B} \sum_{j_A=B}^{C+B} S(C+B, i_B; j_A, j_B) + \sum_{i_B=C+1}^{C+B} \sum_{j_B=i_B-C}^{i_B} \sum_{j_A=0}^{2C-i_B} S(2C-i_B, i_B; j_A, j_B) \\
 & + \sum_{i_A=C}^{C+B-1} \sum_{j_A=i_A-C}^{i_A} \sum_{j_B=0}^{2C-i_A} S(i_A, 2C-i_A; j_A, j_B) + \sum_{i_A=C}^{C+B-1} \sum_{i_B=0}^{2C-i_A-1} \sum_{j_B=0}^{i_B} S(i_A, i_B; i_A-C, j_B)(1-p)
 \end{aligned} \tag{X4}$$

3.5. Descripción del método de simulación en un sistema de dos células.

El programa de simulación que utilizamos para el modelo de dos células está escrito en C. Utiliza el entorno de simulación SMPL.

En el control de acceso de canal tenemos en cuenta las siguientes variables:

- i_A = llamadas en curso en célula A
- i_B = llamadas en curso en célula B
- j_A = llamadas en curso dentro de la zona de load balancing de célula A
- j_B = llamadas en curso dentro de la zona de load balancing de célula B
- C = número de canales en cada célula
- B = número de canales prestables en cada célula
- p = probabilidad de que un nuevo usuario participe en un balance de carga

El control de acceso de canal para una célula A viene determinado según el número de llamadas en curso dentro de la propia célula y de su vecina:

- si $(0 < i_A < C) \ \&\& \ (i_B + i_A < 2C)$ entonces
 - asignamos un canal a la nueva llamada: i_A++
 - si llamada en zona de lb entonces
 - el canal asignado pertenece a la zona de lb: j_A++
- si $(C + B > i_A \ \>= \ C) \ \&\& \ (i_B + i_A < 2C)$ entonces
 - si llamada en zona de lb entonces
 - asignamos un canal a la nueva llamada: i_A++
 - el canal asignado pertenece a la zona de lb: j_A++
 - si no
 - si hay canales propios en zona de lb entonces
 - si directo
 - la llamada se pierde: token_kill
 - si no
 - asignamos un canal a la nueva llamada: i_A++
 - si no
 - la llamada se pierde: token_kill
- si $(i_A = C + B \ \&\& \ i_A + i_B < 2C) \ || \ (i_A + i_B = 2C)$ entonces
 - la llamada se pierde: token_kill

Otra parte importante en el programa es la función *simular* que lleva a cabo la planificación de los eventos. Realiza la extracción de los eventos de la tabla de eventos y luego los procesa. Los más importantes son el evento de una nueva llamada, el evento de un nuevo servicio, el evento fin de servicio y el evento transitorio. Los tres primeros son los necesarios para el procedimiento de servicio de una llamada desde su inicio hasta su terminación. El evento transitorio se utiliza para restablecer los contadores de las llamadas ofrecidas y perdidas hasta un tiempo determinado, para de esta forma obtener unos resultados más fiables. Es importante no restablecer las variables de

estado. Además, la función *simular* contabiliza las llamadas ofrecidas y aceptadas para el cálculo de la probabilidad de pérdidas.

- $PP[i]=(cell_LC[i])/(cell_OC[i])$

Para la función *simular* tendremos en cuenta como variables más importantes:

- $serv_m[i]$, $arrv_m[i]$ = medias de tasa de servicio y tasa de llamadas en célula i
- $serv_s[i]$, $arrv_s[i]$ = desviación estándar de tasa de servicio y tasa de llamadas en célula i
- $serv_d[i]$, $arrv_d[i]$ = tipo de distribución de tasa de servicio y tasa de llamadas en célula i
- sim_duratn = tiempo de simulación
- $cell_OC[i]$ = llamadas ofrecidas en célula i
- $cell_LC[i]$ = llamadas perdidas en célula i
- $cell_AC[i]$ = llamadas aceptadas en célula i
- $next_call_t$ = tiempo de la siguiente llamada
- $call_st$ = duración de la llamada
- $call_lbt$, = duración de residencia en la zona de load balancing
- $call_nolbt$ = duración de residencia fuera de la zona lb
- tk_time_lf = tiempo restante de llamada
- $PP[i]$ = probabilidad de pérdidas en la célula i

Para más información acerca del pseudocódigo del programa de simulación para un sistema de dos células, ver anexos.

3.6. Movilidad en un sistema de dos células.

La movilidad en un sistema de dos células se ha gestionado considerando que la llamada pasa de la zona de load balancing a una zona fuera de load balancing (y viceversa) en la célula donde se inicia la llamada. Es una gestión aproximada de una versión más real que considera una travesía entre células (ver figura 36). No obstante, en el modelo multicelular razonaremos un método para considerar que la llamada atraviesa las células.

Como observamos en el dibujo de la derecha de la figura 36, en la movilidad de dos células consideramos que la llamada nunca sale de la célula donde se origina, si no que la movilidad consiste en trasladar la llamada de la zona de load balancing a la zona fuera de load balancing de la propia célula, mientras la duración de la llamada sea mayor que el tiempo de residencia en cada zona.

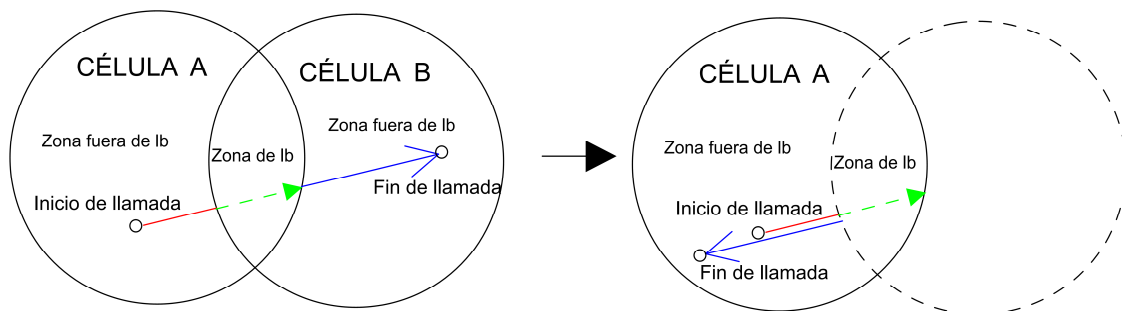


Fig. 36. Ejemplo de movilidad en un sistema de dos células

En concreto en un sistema compuesto por dos células, cuando se presta el servicio de una nueva llamada y existe movilidad, la duración de residencia en la zona de load balancing (o de no load balancing) depende del tipo de distribución del servicio elegida, según sea exponencial, erlang o hyper-exponencial y también depende de la tasa de servicio de la llamada en dicha zona.

Si el usuario está situado en la zona de load balancing y se inicia el servicio de una llamada, la planificación del fin de la llamada depende del tiempo restante de la llamada, es decir, restamos la duración de la llamada menos la duración de residencia en la zona de load balancing. Si este tiempo es mayor que cero entonces se planifica el fin de la llamada para una duración igual a la duración de residencia en la zona de load balancing. En el caso de que sea menor o igual a 0, no existe dicha limitación, y se planifica para la duración de la llamada (ver figura 37).

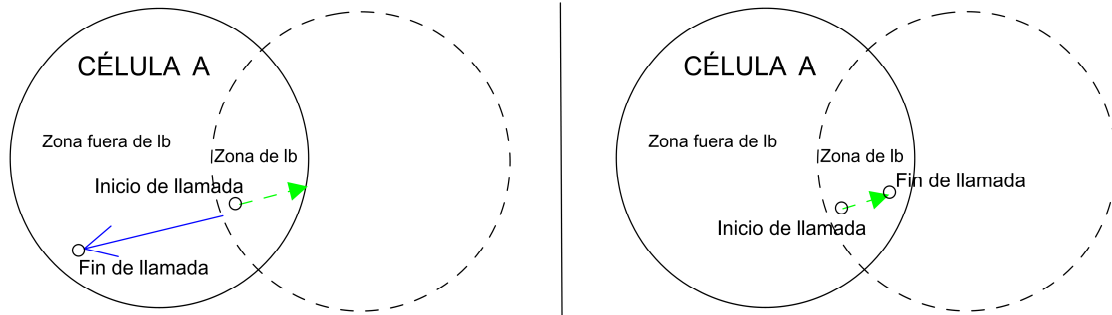


Fig. 37. Ejemplo de movilidad en un sistema de dos células cuando el usuario está situado en la zona de load balancing.

Si el usuario está situado fuera de la zona de load balancing y se inicia el servicio de una llamada, la planificación del fin de la llamada depende del tiempo restante de la llamada, es decir, restamos la duración de la llamada menos la duración de residencia fuera de la zona de load balancing. Si este tiempo es mayor que cero entonces se planifica el fin de la llamada para una duración igual a la duración de residencia fuera de la zona de load balancing. En el caso de que sea menor o igual a 0, no existe dicha limitación, y se planifica para la duración de la llamada (ver figura 38).

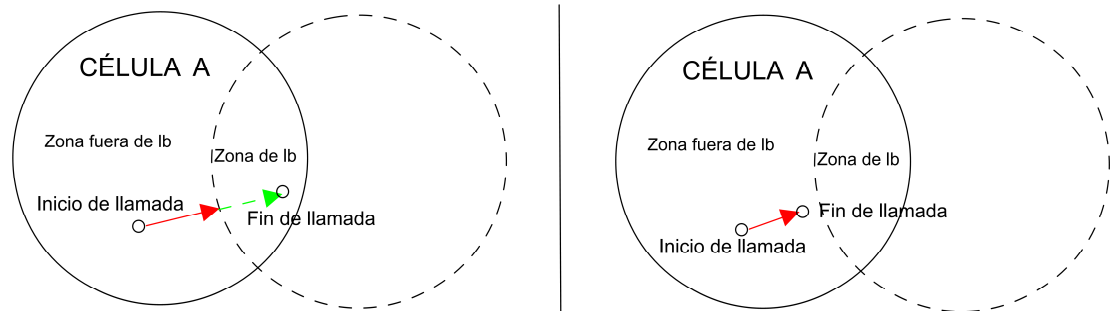


Fig. 38. Ejemplo de movilidad en un sistema de dos células cuando el usuario está situado fuera de la zona de load balancing.

Para determinar cuando finaliza una llamada, tenemos en cuenta el tiempo restante de la llamada tras atravesar una zona de load balancing (o de no load balancing).

Si el tiempo restante de la llamada es mayor que cero y el usuario estaba situado en la zona de load balancing, la planificación del fin de la llamada depende del tiempo que le queda a la llamada, es decir, restamos el tiempo restante de la llamada menos la duración de residencia fuera de la zona de load balancing. Si este tiempo es mayor que cero, entonces se planifica el fin de la llamada para una duración igual a la duración de residencia fuera de la zona de load balancing. En el caso de que sea menor o igual a cero, no existe dicha limitación, y se planifica para la duración restante de la llamada (ver figura 39).

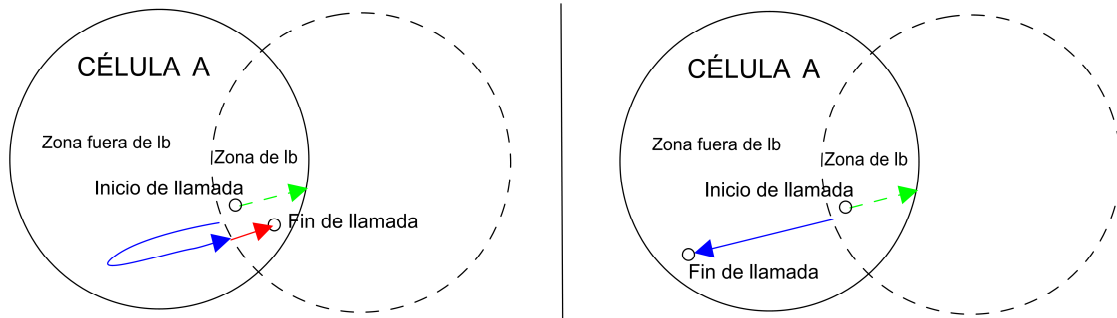


Fig. 39. Ejemplo de movilidad en un sistema de dos células para determinar cuando finaliza la llamada cuando el usuario estaba situado en la zona de load balancing. En el dibujo de la izquierda, el tiempo restante de llamada fuera de la zona de load balancing es mayor que la duración de residencia en esta zona, por lo que la llamada continúa desplazándose. En el dibujo de la derecha, el tiempo restante de llamada fuera de la zona de load balancing es menor que la duración de residencia en esta zona, por lo que se planifica el fin de la llamada para la duración restante de la llamada.

Si el tiempo restante de la llamada es mayor que cero y el usuario estaba situado en la zona fuera de load balancing, la planificación del fin de la llamada depende del tiempo que le queda a la llamada, es decir, restamos el tiempo restante de la llamada menos la duración de residencia de la zona de load balancing. Si este tiempo es mayor que cero, entonces se planifica el fin de la llamada para una duración igual a la duración de residencia de la zona de load balancing. En el caso de que sea menor o igual a cero, no existe dicha limitación, y se planifica para la duración restante de la llamada (ver figura 40).

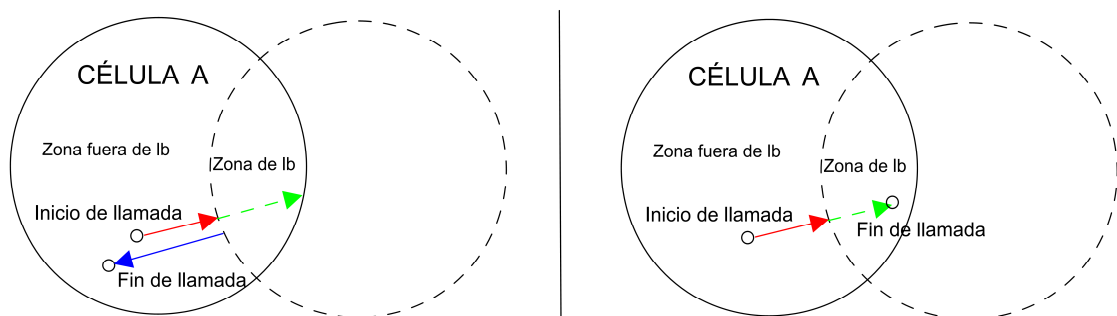


Fig. 40. Ejemplo de movilidad en un sistema de dos células para determinar cuando finaliza la llamada cuando el usuario estaba situado fuera de la zona de load balancing. En el dibujo de la izquierda, el tiempo restante de llamada en la zona de load balancing es mayor que la duración de residencia en esta zona, por lo que la llamada continúa desplazándose. En el dibujo de la derecha, el tiempo restante de llamada en la zona de load balancing es menor que la duración de residencia en esta zona, por lo que se planifica el fin de la llamada para la duración restante de la llamada.

Cuando existe movilidad en un sistema compuesto por dos células, se reduce la probabilidad de pérdidas, como veremos en la representación gráfica.

3.7. Resultados y simulación gráfica en un sistema de dos células.

A continuación queremos comparar de forma gráfica los resultados obtenidos mediante los modelos analíticos y nuestro programa de simulación para los métodos de balance de carga directo e indirecto en el caso de un sistema de dos células sin movilidad. Los modelos analíticos están resueltos mediante unos programas en Matlab y los modelos de simulación mediante un programa en C y las librerías SMPL.

En la siguiente tabla se detallan los parámetros usados en nuestro programa de simulación.

<i>Posición del vector</i>	<i>Parámetros</i>	<i>Descripción</i>
1	C	Número de canales de cada célula
2	B	Número de canales prestables
3	P	Probabilidad de que un nuevo usuario participe en un balance de carga
4	μ_A	Tasa de servicio en la célula A, $E[s_A] = 1/\mu_A$
5	$\sigma(s_A)$	Desviación típica de s_A
6	Distribución de s_A	1=exponencial, 2=erlang, ó 3=hyper-exponencial
7	μ_B	Tasa de servicio en la célula B, $E[s_B] = 1/\mu_B$
8	$\sigma(s_B)$	Desviación típica de s_B
9	Distribución de s_B	1=exponencial, 2=erlang, ó 3=hyper-exponencial
10	λ_A	Tasa de llegada en la célula A, $E[\tau_A] = 1/\lambda_A$
11	$\sigma(\tau_A)$	Desviación típica de τ_A
12	Distribución de τ_A	1=exponencial, 2=erlang, ó 3=hyper-exponencial
13	λ_B	Tasa de servicio en la célula B, $E[\tau_B] = 1/\lambda_B$
14	$\sigma(\tau_B)$	Desviación típica de τ_B
15	Distribución de τ_B	1=exponencial, 2=erlang, ó 3=hyper-exponencial
16	Tiempo de simulación	0 a infinito
17	Número de simulaciones	Simulaciones por línea de parámetros: 1-15
18	Porcentaje de transitorio	0% - 100%
19	MOBTY	0: NO, 1: YES
20	DIRECT	1: Método directo, 0: Método indirecto

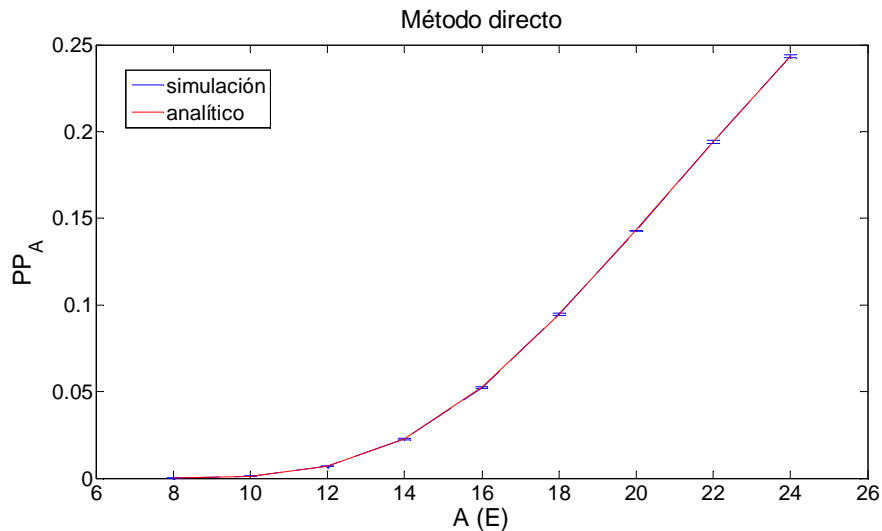
Para ello hemos tomado unos valores ejemplo para los parámetros de nuestro sistema bicelular. A continuación se muestran los valores que hemos mantenido fijos en nuestro ejemplo.

<i>Parámetros de nuestro ejemplo</i>	
C	20
B	6
P	0.5
μ_A	1
$\sigma (s_A)$	1
Distribución de s_A	Exponencial
μ_B	1
$\sigma (s_B)$	1
Distribución de s_B	Exponencial
λ_A	-
$\sigma (\tau_A)$	1
Distribución de τ_A	Exponencial
λ_B	-
$\sigma (\tau_B)$	1
Distribución de τ_B	Exponencial
Tiempo de simulación	50000
Número de simulaciones	5
Porcentaje de transitorio	10%
MOBTY	-
DIRECT	-

Tomamos 5 muestras de los resultados obtenidos con nuestro programa de simulación, es decir, 5 simulaciones por punto, y calculamos su media. Esta media será el valor que utilizemos para cada punto de nuestra gráfica. Además para cada punto representamos los intervalos de confianza del 95%, es decir, los intervalos donde la media real se encuentra con una probabilidad del 95%.

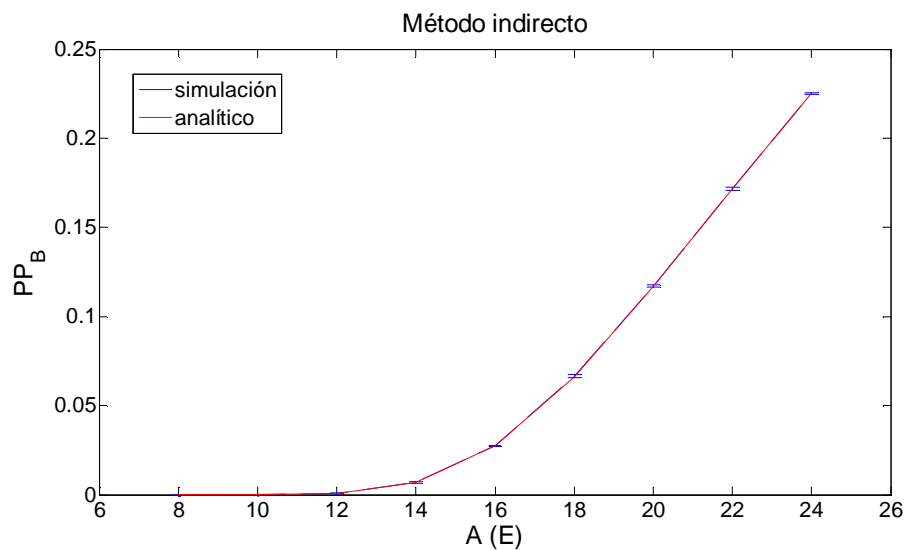
Representamos los valores obtenidos tanto en el método analítico como en la simulación y observamos la gran fiabilidad de nuestro programa de simulación, puesto que los puntos de nuestro modelo analítico se encuentran dentro de los intervalos de confianza.

La gráfica 1 representa la probabilidad de pérdidas para el método directo en la célula A (PP_A) de un sistema de dos células en el que hacemos variar el tráfico ofrecido (A) por ambas células, expresado en Erlangs (E). En la gráfica 1 vemos que la probabilidad de pérdidas en la célula A para el método directo aumenta con el tráfico ofrecido en cada una de las dos células de forma creciente y con una pendiente cada vez mayor antes de la saturación.



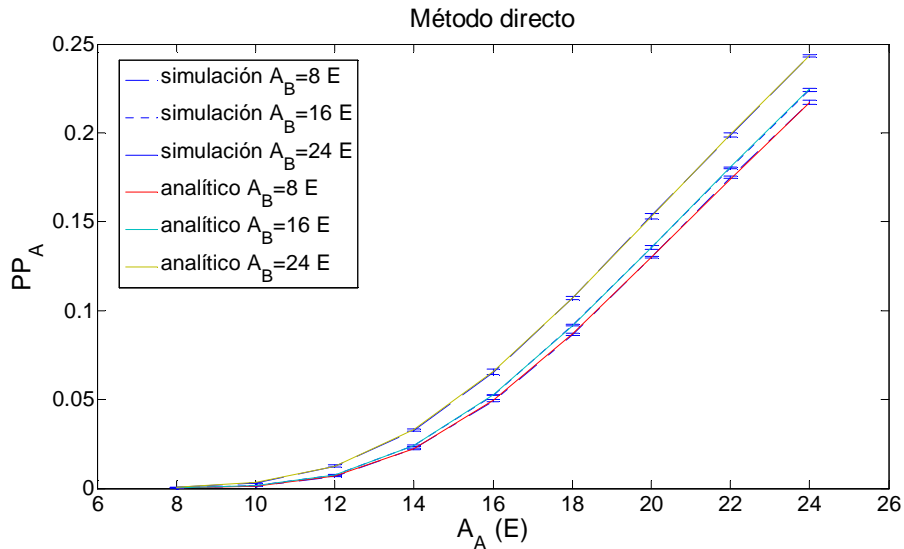
Gráf. 1. PP_A / A para el método directo sin movilidad y mismo A en ambas células.

La gráfica 2 representa el mismo caso que la gráfica 1 para el método indirecto. En la gráfica 2 ocurre lo mismo que en la gráfica 1 sin embargo para el método indirecto la pendiente de la probabilidad de pérdidas es menor que para en el método directo.



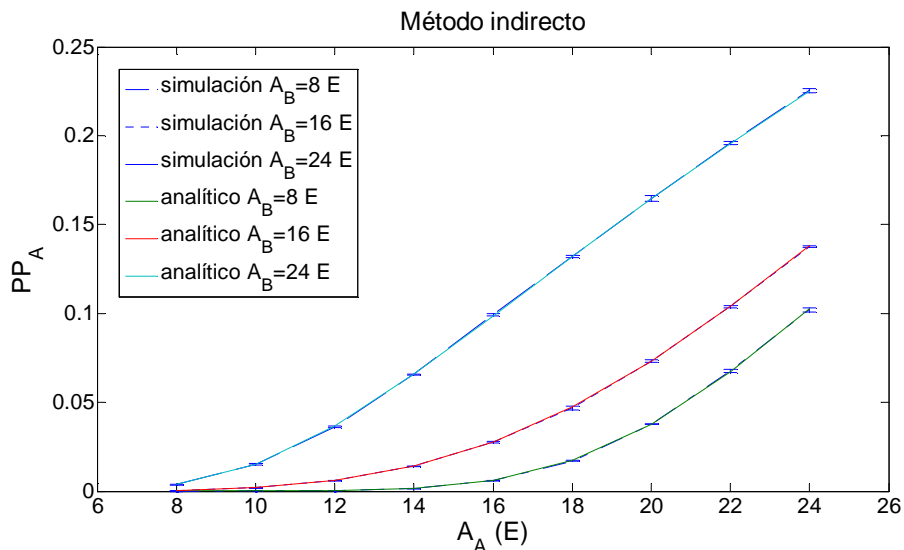
Gráf. 2. PP_A / A para el método indirecto sin movilidad y mismo A en ambas células.

La gráfica 3 representa las probabilidades de pérdidas para el método directo en la célula A (PP_A) de un sistema de dos células en el que hacemos variar el tráfico ofrecido por la célula A (A_A), y mantenemos fijo el tráfico ofrecido por la célula B, para $A_B = 8 E$, $A_B = 16 E$, $A_B = 24 E$. En la gráfica 3, observamos para el método directo que dado un tráfico fijo en la célula B, la probabilidad de pérdidas en la célula A es mayor conforme el tráfico en B es mayor. Además el incremento en cada punto es proporcionalmente cada vez mayor conforme el tráfico en B es mayor.



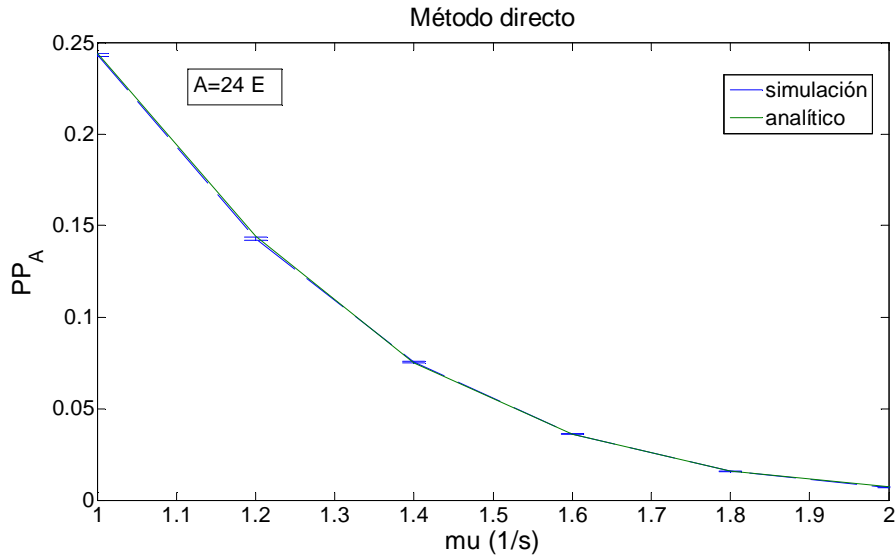
Gráf. 3. PP_A / A_A para el método directo sin movilidad y para A_B fijo.

La gráfica 4 representa el mismo caso que la gráfica 3 para el método indirecto. En la gráfica 4, observamos un comportamiento parecido para el método indirecto. El aumento proporcional para cada curva es mayor que en el método directo.



Gráf. 4. PP_A / A_A para el método indirecto sin movilidad y para A_B fijo.

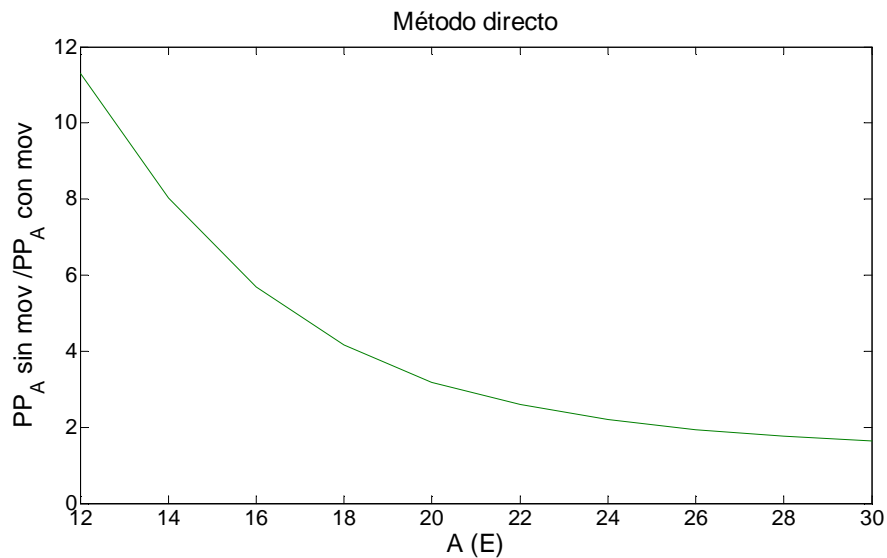
La gráfica 5 representa la probabilidad de pérdidas para el método directo en la célula A (PP_A) de un sistema de dos células en el que hacemos variar la tasa de servicio de por ambas células, manteniendo fijo el tráfico ofrecido en ambas células ($A = 24$ E). En la gráfica 5, vemos que para un tráfico ofrecido fijo en ambas células, a medida que la tasa de servicio es mayor, la probabilidad de pérdidas en la célula A es menor.



Gráf. 5. PP_A / μ para el método directo sin movilidad y mismo A en ambas células.

A continuación tenemos en cuenta la movilidad y representamos varias gráficas para observar su rendimiento frente a un sistema sin movilidad.

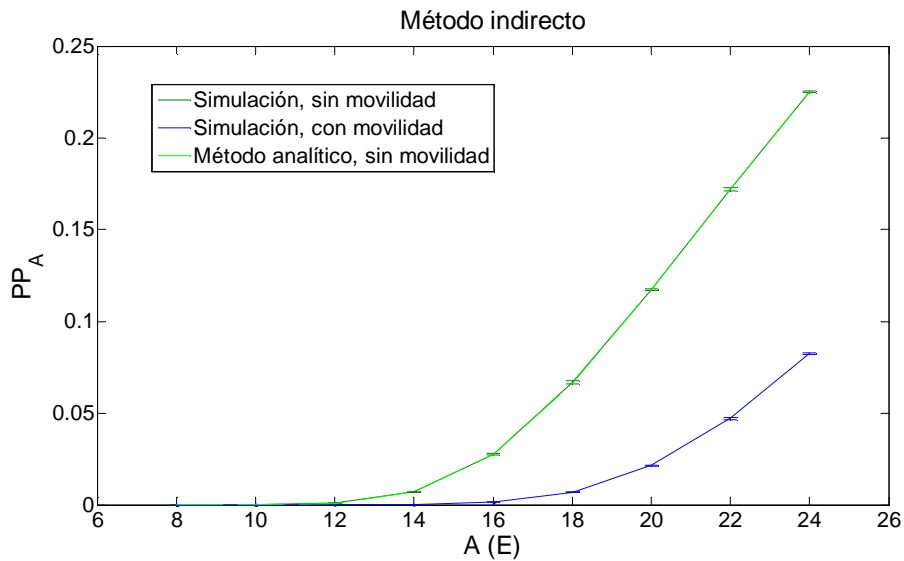
La gráfica 6 representa el cociente de la probabilidad de pérdidas en la célula A en un sistema sin movilidad y la probabilidad de pérdidas en la célula A en un sistema de dos células con movilidad para el método directo en el que hacemos variar el tráfico ofrecido (A) por ambas células, expresado en Erlangs (E). En la gráfica 6, notamos que la probabilidad de pérdidas sin movilidad es mayor que la probabilidad de pérdidas con movilidad. Conforme el tráfico ofrecido es mayor, la diferencia proporcional entre ambas probabilidades es cada vez menor.



Gráf. 6. $PP_A \text{ sin movilidad} / PP_A \text{ con movilidad}$ para el método directo y mismo A en ambas células.

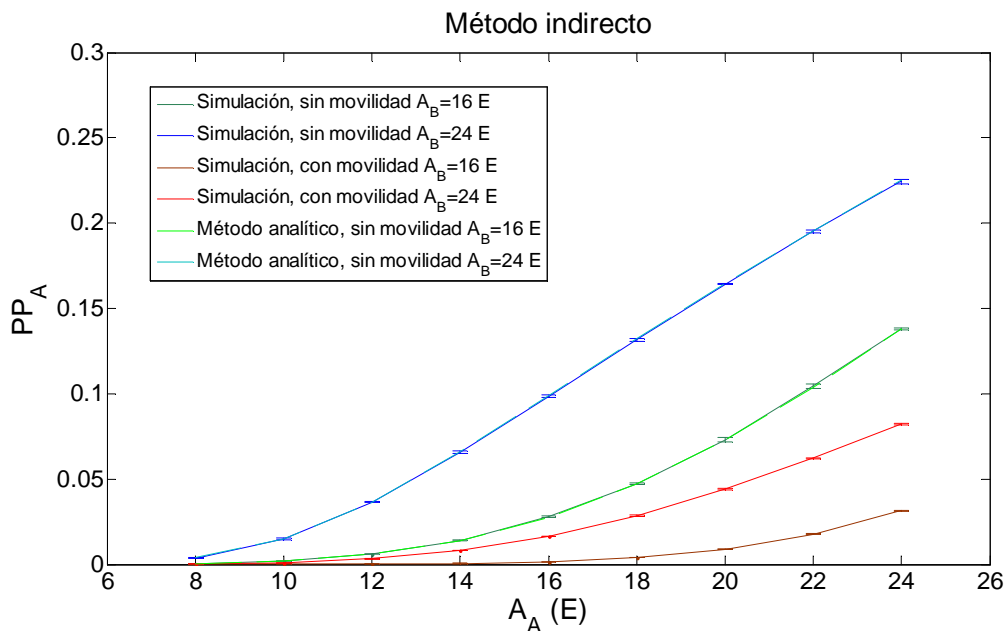
La gráfica 7 representa las probabilidades de pérdidas para el método indirecto en la célula A (PP_A) de un sistema de dos células cuando hay y no hay movilidad en el que hacemos variar el tráfico ofrecido (A) por ambas células, expresado en Erlangs (E). En la gráfica 7, observamos como la probabilidad de pérdidas sin movilidad es cada vez

mayor a la probabilidad de pérdidas con movilidad conforme el tráfico ofrecido (A) es mayor.



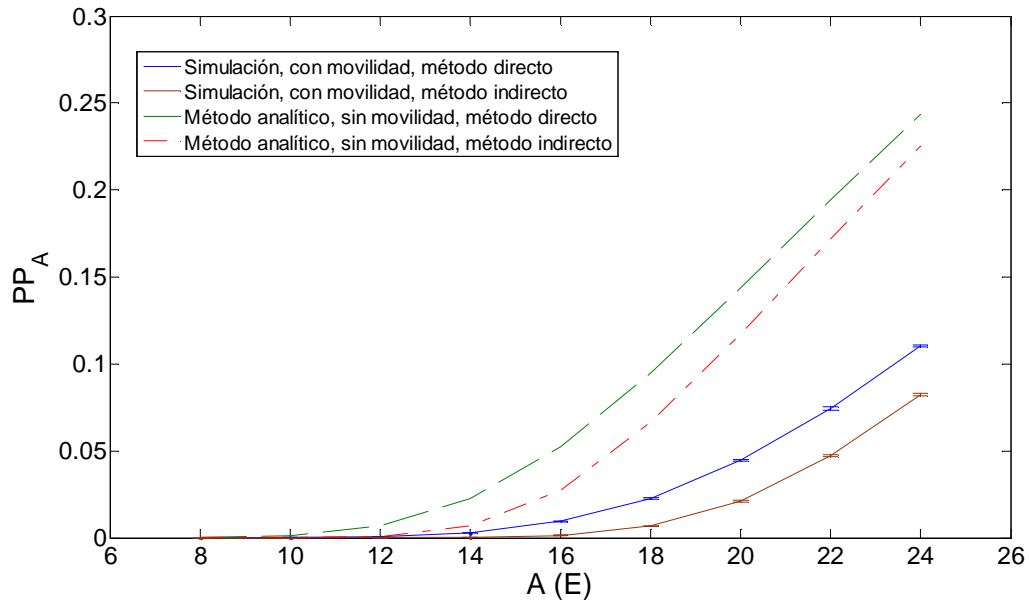
Gráf. 7. PP_A / A para el método directo y mismo A en ambas células, con y sin movilidad.

La gráfica 8 representa las probabilidades de pérdidas para el método indirecto en la célula A (PP_A) de un sistema de dos células con y sin movilidad en el que hacemos variar el tráfico ofrecido por la célula A (A_A), y mantenemos fijo el tráfico ofrecido por la célula B, para $A_B = 8$ E, $A_B = 16$ E, $A_B = 24$ E. En la gráfica 8, vemos que la probabilidad de pérdidas para sistemas sin movilidad para un tráfico fijo en la célula B es mayor que cuando hay movilidad. Para un tráfico fijo en la célula B mayor, la probabilidad de pérdidas en A es cada vez mayor. La pendiente para un tráfico de 16 E es mayor que para un tráfico de 24 E. La diferencia de la probabilidad de pérdidas para tráficos fijos de 16 y 24 E cuando no hay movilidad es mayor que cuando hay movilidad.



Gráf. 8. PP_A / A_A para el método indirecto y para A_B fijo, con y sin movilidad.

La gráfica 9 representa las probabilidades de pérdidas para el método directo e indirecto en la célula A (PP_A) de un sistema de dos células cuando hay y no hay movilidad en el que hacemos variar el tráfico ofrecido (A) por ambas células, expresado en Erlangs (E). En la gráfica 9, notamos que la probabilidad de pérdidas cuando no hay movilidad es mayor que cuando hay movilidad y que la probabilidad de pérdidas en el método directo es mayor que en el método indirecto. Además, la pendiente cuando no hay movilidad es mayor que cuando la hay.



Gráf. 9. PP_A / A para el método directo e indirecto, mismo A en ambas células, con y sin movilidad.

4. Los métodos de balance de carga en un sistema multicelular

Las propuestas que se han planteado para el balance de carga en redes celulares que hemos estudiado de la documentación no tienen en cuenta la movilidad. Por tanto, nuestra intención es hacer una simulación de la probabilidad de bloqueo en una célula de un sistema multicelular real cuando existe movilidad con la mayor precisión posible.

El sistema multicelular que se ha implementado es un escenario de 3 capas en forma de anillos. El efecto del borde de este escenario se considera despreciable, es decir, aunque las células del tercer anillo tengan un número de vecinas menor que el resto de los anillos, los resultados que obtenemos para la célula central siguen siendo fiables puesto que esta célula se considera suficientemente aislada de la tercera capa (ver figura 41).

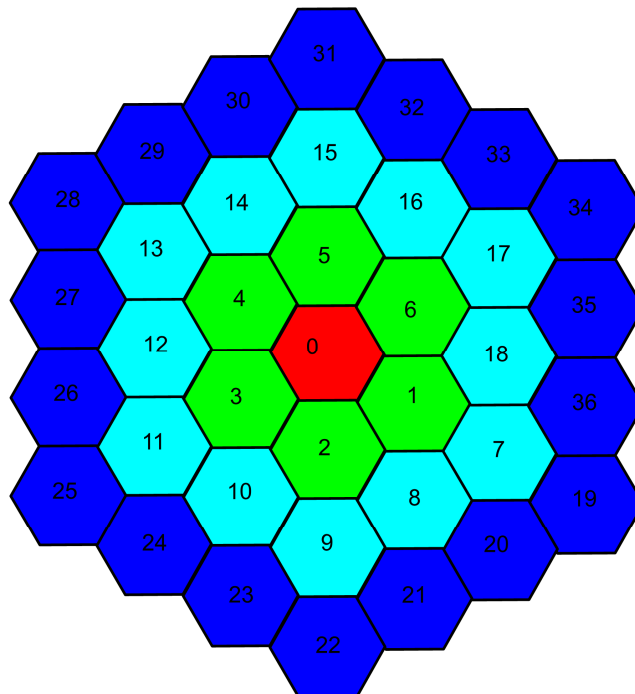


Fig. 41. Ejemplo de un sistema multicelular de 3 anillos compuesto por múltiples *clusters*.

El modelo de simulación que hemos planteado para el método directo y para el método indirecto en un sistema multicelular requiere un estudio por fases, es decir, no se puede realizar directamente. Primeramente, no podemos plantear la movilidad del sistema sin previamente haber considerado el sistema estático sin movilidad. En segundo lugar, el análisis de este sistema tan grande de forma individual para cada célula mediante un método matemático tampoco es posible de forma directa ya que requiere muchas dimensiones para nuestros procesos de Markov, además de mucha carga computacional.

Para ello debemos plantear un escenario sencillo, por ejemplo que nuestro sistema multicelular sólo tiene 3 células útiles. Consideramos que únicamente se originan llamadas y se pueden pedir canales en 3 células. Si los resultados del programa de simulación coinciden con el modelo analítico sencillo de 3 células que sí podemos

realizar, estamos seguros de que nuestro programa de simulación del sistema multicelular funciona para un mayor número de células.

El modelo analítico de 3 células se puede realizar con facilidad, puesto que el número de estados final del modelo es proporcional a una potencia al cubo del número de canales que tiene cada célula. Para 10 canales, el número de estados es 1000, lo cual es el límite para el funcionamiento correcto de Matlab. Para mayor número de células y canales el problema es inabordable.

Una vez hemos realizado el modelo analítico de 3 células, el siguiente paso es la implementación del programa de simulación para el sistema multicelular, que se basa en un conjunto de *clusters* interconectados. La configuración de un sistema multicelular compuesto por 3 anillos se basa en considerar cada célula como la célula central de un clúster y sabiendo cuales son los vecinos de cada una de las células podemos calcular la probabilidad de pérdidas del sistema global.

De esta forma el trabajo fundamental será el estudio de un *cluster* de 7 células, que consiste en una *hot cell* y sus vecinos. Un estudio analítico de un *cluster* de 7 células para los métodos directo e indirecto es prácticamente irrealizable. Sin embargo, la dinámica de un *cluster* de un menor número de células, por ejemplo de 3 células, y de un *cluster* de 7 células es la misma.

Por consiguiente, realizamos un estudio de un *cluster* compuesto por 3 células, que resulta mucho más sencillo de implementar analíticamente que un *cluster* de 7 células, en el cual la célula central representa el *hot spot*. La manera de realizar un modelo analítico de un *cluster* de 3 células es similar al realizado en un sistema de 3 células, pero en este caso la célula central es capaz de pedir prestados canales de sus vecinas y sus vecinas pueden pedir prestados canales de ella, pero no entre sí (en el caso del sistema de 3 células, todas las células pueden pedir canales entre sí mismas).

Suponemos que si nuestro programa de simulación funciona para un clúster de 3 células, también funciona para un clúster de 7, es decir, si los resultados de la simulación del *cluster* de 3 células son aproximados a los de nuestro sistema analítico realizado en Matlab, podemos extrapolar los resultados y declarar que nuestro programa de simulación sirve para un *cluster* de 7 células compuesto una célula central y sus vecinas. La diferencia en el programa de simulación para un clúster de 7 células y el de 3 células es únicamente un parámetro que indica el número de células que constituyen el *cluster*, pero su funcionamiento es exactamente el mismo.

La figura 42 representa un *cluster* de 3 células de un sistema multicelular con la TTR en el área fronteriza de las células A, B, y C. Consideramos una $p = 0.05$. Para la célula central contamos con 2 células vecinas, por tanto la probabilidad de load balancing total en esta célula es $p = 0.1$. Para las células exteriores, $p = 0.05$.

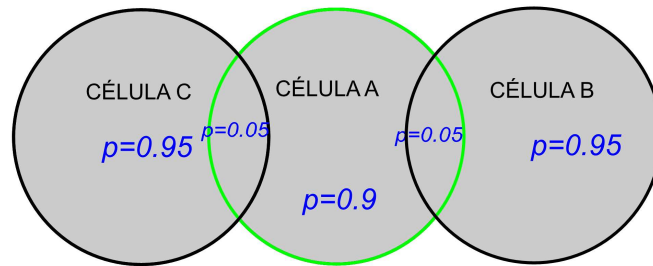


Fig. 42. Ejemplo de un *cluster* de 3 células con TTR en la zona fronteriza.

A continuación presentamos en la figura 43, a modo de ejemplo, dos visiones de un *cluster* de un sistema multicelular con la TTR en el área fronteriza de las células, con $C=3$ canales propios de cada célula y $B=2$ canales que se pueden usar para balance de carga en cada célula, $i_k = 2$ llamadas en curso en la célula k , donde $k = A, B, C, D, E, F, G$. Consideramos un *cluster* de 7 células y una $p = 0.05$. Para la célula central contamos con 6 células vecinas, por tanto la probabilidad de load balancing en esta célula es $p = 0.3$. Para las células exteriores, $p = 0.05$.

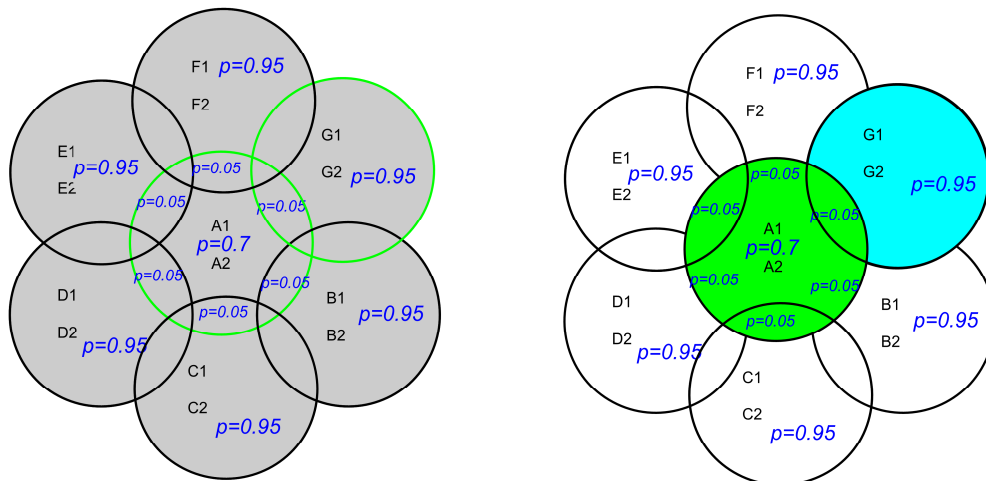


Fig. 43. Ejemplo de un *cluster* de 7 células con TTR en la zona fronteriza.

Una vez hemos demostrado que los resultados de nuestro programa de simulación para el modelo multicelular se ajustan a los del modelo analítico, podemos proceder a hacer un estudio de modelos de células más complejos con todas las células operativas que utilizan más canales, es decir, aquellos en los que todas las células pueden ser origen de las llamadas y puedan pedir prestados canales.

4.1. Descripción del modelo analítico en un sistema multicelular.

El método analítico en un sistema multicelular estudiado en [1] se basa en el estudio de un modelo matemático más sencillo que en un sistema de dos células.

El análisis exacto de un sistema práctico de múltiples *hot spots* es muy difícil de evaluar. Por eso en [1] se realiza un análisis aproximado, teniendo como base un sistema de dos células.

El sistema lo divide en *clusters*, que consisten en una *hot cell* y sus vecinos, en lugar de estudiar cada célula individualmente (ver figura 44). Cada *hot cell* está situada en el centro de cada *cluster* y es capaz de pedir prestados canales a sus vecinos dentro de cada *cluster*. [1] considera cada *cluster* como un sistema de dos células con una *hot cell* y una *cold cell*, la cual se corresponde con las 6 células vecinas de la *hot cell*.

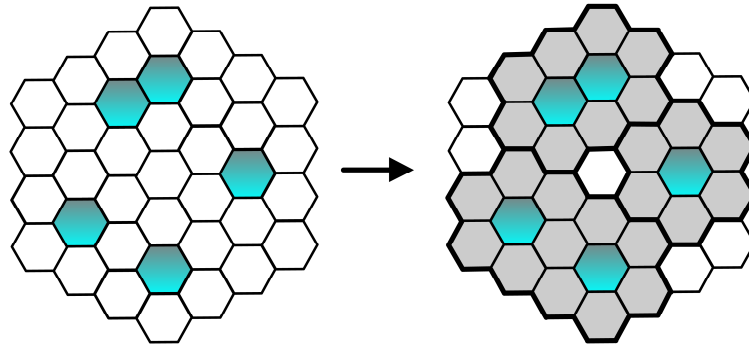


Fig.44. Ejemplo de un sistema multicelular y los correspondientes *clusters* del sistema.

El método de resolución que se estudia en [1] define una variable llamada “probabilidad de bloqueo en las células vecinas”, en lugar de tener en cuenta todos los canales ocupados en las células vecinas, lo cual facilita el análisis del sistema multicelular.

Asume llegadas de Poisson y tiempos de servicio exponencialmente distribuidos. Con ello puede describir el número de llamadas en curso mediante un proceso de Markov [10] [11]. Los procesos de Markov obtenidos serán unidimensionales o bidimensionales, según trabajemos con el método directo o indirecto, respectivamente.

Los resultados que obtenemos del sistema multicelular a partir del modelo analítico y mediante la probabilidad de bloqueo en las células vecinas que se ha trabajado en [1], no nos permitirán hacer una comparación con los resultados obtenidos del modelo de simulación. La razón es que la variable utilizada que representa la probabilidad de bloqueo en las células vecinas, es una variable externa totalmente incorrelada que no se puede considerar como una variable de entrada en nuestro sistema de simulación, si no que nuestro programa de simulación obtiene la probabilidad de bloqueo a partir de las llamadas que se pierden del total de llamadas que intentan iniciarse. Por lo tanto hemos buscado otra solución para constatar que nuestro programa de simulación funciona correctamente.

4.1.1. Descripción del modelo analítico en un sistema multicelular para el método directo.

Para el método directo estudiado en [1], los estados del proceso de Markov serán unidimensionales ya que en un sistema multicelular el tráfico de las células vecinas hacia las *hot cells* se considera despreciable. Los estados del proceso de Markov se definen como $S(i)$, donde i es el número de llamadas en curso en la célula bajo estudio.

El tamaño de los procesos de Markov (ver figura 45) para el método directo en un sistema multicelular depende de varios parámetros, como son el número de canales de cada célula, C , el número de canales prestables, B , la probabilidad de que un nuevo usuario participe en un balance de carga, p , la tasa de llegadas de la célula, λ , la tasa de servicio de la célula μ , y la probabilidad de bloqueo en las células vecinas, P_b .

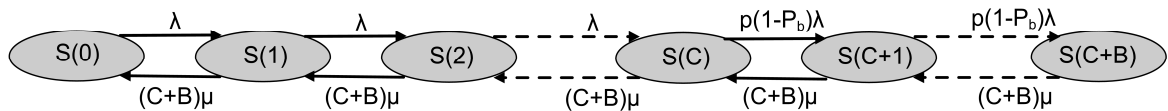


Fig.45. Diagrama de estados para balance de carga directo para un sistema multicelular.

La forma de calcular las probabilidades de estado y la probabilidad de pérdidas del sistema será inmediata pues se trata de un sistema unidimensional. Para ello, realizaremos un programa en Matlab para el método directo (y otro programa para el método indirecto) que tenga en cuenta la propiedad de reversibilidad de los procesos de Markov.

Una vez hemos obtenido las probabilidades de estado, determinamos cuando una nueva petición de llamada será bloqueada en nuestro sistema:

- Una nueva llamada en la célula será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + B$, están siendo utilizados.
- Una nueva llamada será bloqueada con probabilidad $(1 - p - pP_b)$ si
 - Todos los canales de la célula se utilizan y, aunque hay canales disponibles para ser prestados, la nueva llamada
 - No puede participar en el balance de carga.
 - Puede participar en el balance de carga, pero es bloqueada por las células vecinas.

El siguiente paso consiste en determinar cuales son los eventos de bloqueo de llamada en la célula y a qué estados corresponden:

- $S(C+B)$.
- $S(i)$, donde $C \leq i < C + B$.

Por último determinamos la probabilidad de bloqueo de llamada en la célula, sumando los estados de bloqueo anteriormente descritos (ver X5):

$$P_{B_D} = S(C + B) + \sum_{i=C}^{C+B-1} S(i)(1 - p + pP_b) \quad (X5)$$

4.1.2. Descripción del modelo analítico en un sistema multicelular para el método indirecto.

Para el método indirecto estudiado en [1], los estados del proceso de Markov se definen como $S(i; j)$, donde i es el número de llamadas en curso en la célula bajo estudio, y j el número de llamadas en curso en la zona de load balancing de la célula. Debemos considerar el número de llamadas en curso que pueden usarse para load balancing, en el caso en que los nuevos usuarios no puedan participar en load balancing directamente.

El tamaño y forma de los procesos de Markov (ver figura 46) para el método indirecto en un sistema multicelular también depende de los mismos parámetros que el método directo, como son C , B , p , λ , μ y P_b .

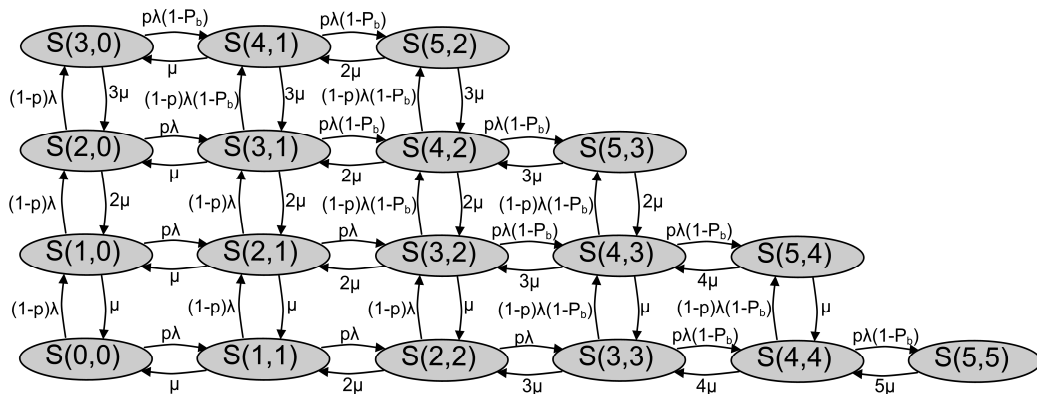


Fig.46. Diagrama de estados en un sistema multicelular ejemplo con $C = 3$ y $B = 2$ en el balance de carga indirecto.

Mediante los procesos de Markov podemos calcular las probabilidades de estado y la probabilidad de pérdidas de los sistemas. La forma de resolverlos se puede hacer mediante las ecuaciones de balance global o teniendo en cuenta que los procesos son reversibles. En este caso hemos optado por resolverlos mediante las ecuaciones de balance global, pues la composición del proceso no facilita la resolución del proceso reversible de una manera sencilla.

Una vez hemos obtenido las probabilidades de estado, determinamos cuando una nueva petición de llamada será bloqueada en nuestro sistema:

- Una nueva llamada en la célula será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + B$, están siendo utilizados

- Una nueva llamada será bloqueada con probabilidad $(1 - p + pP_b)$ si
 - Todos los canales de la célula se utilizan, no hay llamadas en curso para emplear el balance de carga indirecto y la nueva llamada
 - No puede participar en el balance de carga directamente.
 - Puede participar en el balance de carga, pero es bloqueada por las células vecinas.
- Una nueva llamada será bloqueada con probabilidad P_b si
 - Todos los canales de la célula se utilizan y al menos hay una llamada en curso para emplear el balance de carga dinámico, pero la nueva llamada es bloqueada por las células vecinas.

El siguiente paso consiste en determinar cuales son los eventos de bloqueo de llamada en la célula y a qué estados corresponden:

- $S(C+B; j)$ donde $i - C \leq j \leq i$
- $S(i; i - C)$, donde $C \leq i < C + B$
- $S(i; j)$, donde $C \leq i < C + B, i - C + 1 \leq j \leq i$

Por último determinamos la probabilidad de bloqueo de llamada en la célula, sumando los estados de bloqueo anteriormente descritos (ver X6):

$$P_{B_{ID}} = \sum_{j=B}^{C+B} S(C+B; j) + \sum_{i=C}^{C+B-1} S(i; i-C)(1-p+pP_b) + \sum_{i=C}^{C+B-1} \sum_{j=i-C+1}^i S(i; j) \quad (X6)$$

4.1.3. Modelo analítico desarrollado para los métodos de balance de carga en un sistema multicelular.

El modelo analítico que hemos desarrollado para el sistema multicelular se basa en realizar un método analítico de 3 células para comprobar el funcionamiento del programa de simulación.

El método analítico para un sistema de 3 células considera que todas las células se intercambian canales. En el método analítico implementado para un *cluster* de 3 células, las células externas sólo intercambian canales con la central, pero no entre sí. La explicación del método analítico de un *cluster* no va a ser un objetivo para este proyecto, aunque presenta muchas similitudes al método analítico del sistema de 3 células.

Asumimos llegadas de Poisson y tiempos exponencialmente distribuidos, con lo que podemos describir el número de llamadas en curso mediante un proceso de Markov [10] [11]. Respecto al método analítico del sistema de 2 células, el número de dimensiones del proceso de Markov se incrementa, puesto que contamos con más zonas posibles donde se puede iniciar la llamada, pasamos de tener una zona de load balancing y dos zonas fuera de la zona de load balancing en el sistema de dos células, a tener tres zonas de load balancing y tres zonas fuera de la zona de load balancing.

Es importante saber en qué zona se inicia la llamada, es decir, si se inicia fuera o dentro de la zona de load balancing. Existe una zona de load balancing por cada célula vecina de la célula donde se inicia la llamada, es decir 2 zonas. Suponemos que p es la fracción de superficie de 2 células en la que se puede hacer load balancing. Si la llamada se inicia en una zona de load balancing, debemos saber con cuál célula vecina se comparte. Para el método indirecto, debemos saber si la llamada situada en la zona de load balancing es una llamada que utiliza un canal propio de la célula o un canal prestado de la célula vecina.

Al igual que sucedía con el modelo de dos células, mediante los procesos de Markov podemos calcular las probabilidades de estado. La forma de obtener dichas probabilidades se hace mediante un método recursivo, ya que los procesos son reversibles.

Para calcular la probabilidad de pérdidas, determinamos cuando una llamada será bloqueada, que dependerá del número de canales utilizados y prestados y de la zona del *cluster* donde se inicie la llamada.

Para el método directo:

- Una nueva llamada en la célula A será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + 2B$, están siendo utilizados.
 - Todos los canales de las células A, B y C, $3C$, están siendo utilizados.

- Una nueva llamada será bloqueada con probabilidad $(1 - 2p)$ ó $(1 - p)$ si
 - Todos los canales de la célula A se utilizan y, aunque hay canales disponibles para ser prestados, el nuevo usuario no puede participar en el balance de carga, debido a su localización.

Para el método indirecto:

- Una nueva llamada en la célula A será bloqueada con probabilidad 1 si
 - El número máximo de canales usados en una célula, $C + 2B$, están siendo utilizados
 - Todos los canales de las células A, B y C, $3C$, están siendo utilizados.
- Una nueva llamada será bloqueada con probabilidad $(1 - 2p)$ ó $(1 - p)$ si
 - Todos los canales de la célula A se utilizan y, aunque hay canales disponibles para ser prestados, no hay llamadas activas para emplear en load balancing.

El siguiente paso consiste en determinar cuáles son los eventos de bloqueo de llamada en la célula A y a qué estados corresponden para el método directo y el método indirecto. Por último determinamos la probabilidad de bloqueo de llamada en la célula A, sumando los estados de bloqueo.

A continuación planteamos un modelo analítico ejemplo de un sistema de 3 células con $C=2$ y $B=1$ (ver figura 47). Para este sistema de 3 células, el método directo tiene en cuenta 3 variables por célula y el método indirecto 5 variables por célula, para contabilizar el número de estados, previo al cálculo de la probabilidad de bloqueo. Consideramos ijk , $ifjk$, $idjk$, el número de llamadas iniciadas en cada una de las zonas del sistema (j y k son números enteros cuyo valor depende de la zona en que se encuentren las llamadas)

Para el método directo, los estados dependen del número de llamadas que están en curso en cada una de las 3 zonas por célula del sistema:

- $st(i_{10}, i_{12}, i_{13}, i_{20}, i_{21}, i_{23}, i_{30}, i_{31}, i_{32})$

Para el método indirecto, los estados dependen del número de llamadas que están en curso en cada una de las zonas del sistema y del tipo de canal utilizado (propio o prestado), en total 5 variables por célula:

- $st(if_{10}, i_{12}, i_{13}, id_{12}, id_{13}, if_{20}, i_{21}, i_{23}, id_{21}, id_{23}, if_{30}, i_{31}, i_{32}, id_{31}, id_{32})$

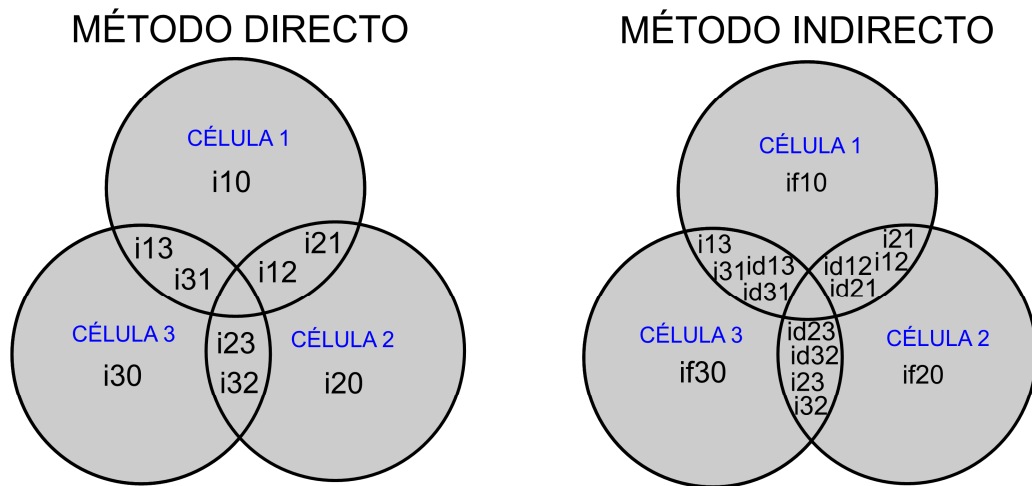


Fig. 47. Sistema de 3 células para el método analítico.

Para el método directo las variables de estado son:

- Las llamadas que se inician fuera de la zona de load balancing de cada célula: i_{10} , i_{20} , i_{30} .
- Las llamadas que se inician en la zona de load balancing común a una célula vecina: i_{12} , i_{13} , i_{21} , i_{23} , i_{31} , i_{32} .

Para el método indirecto las variables de estado son:

- Las llamadas que se inician fuera de la zona de load balancing de cada célula: if_{10} , if_{20} , if_{30} .
- Las llamadas que se inician en la zona de load balancing común a una célula vecina y que utilizan canales propios de la célula: id_{12} , id_{13} , id_{21} , id_{23} , id_{31} , id_{32} .
- Las llamadas que se inician en la zona de load balancing común a una célula vecina y que utilizan canales prestados de una célula vecina: i_{12} , i_{13} , i_{21} , i_{23} , i_{31} , i_{32} .

En el modelo analítico para el método directo con $C=2$, y $B=1$, se obtienen un total de 512 estados. En el modelo analítico para el método indirecto con $C=2$, y $B=1$, se obtienen un total de 6859 estados. Matlab es incapaz de procesar este número de estados y en el cálculo de la probabilidad de bloqueo se produce un error por falta de memoria.

4.2. Descripción del método de simulación en un sistema multicelular compuesto por 3 anillos.

La descripción del método de simulación en un sistema multicelular compuesto por una malla de 3 anillos, es similar a la de un *cluster* de 7 células, añadiendo algunos aspectos importantes. Primero consideramos el método de simulación para el modelo de un *cluster* y luego comentaremos las diferencias fundamentales con el sistema multicelular.

En el control de acceso de canal para un *cluster* tenemos en cuenta las siguientes variables:

- ve = Célula vecina elegida
- $vebuscado$ = Célula vecina cuya zona de load balancing tiene un canal que la célula actual puede adquirir cuando se inicia una nueva llamada
- $is[i]$ = Número de llamadas en curso en la célula i (ya sean canales propios o prestados de otras células)
- $isp[i]$ = Número de llamadas en curso que utilizan canales propios de la célula i
- $js[i]$ = Número de llamadas en curso en zona lb de la célula i (ya sean propios o prestados de otras células)
- $jsz[i][ve]$ = Número de llamadas en curso en zona lb de la célula i que comparte la zona de lb de la célula ve (ya sean propios o prestados de ve)
- $jjs[i][ve]$ = Número de canales que la célula i toma prestados de la célula ve
- $bt[i]$ = Numero de canales utilizados en la célula i ya sean canales propios o canales que está prestando a otras células
- C = número de canales en cada célula
- B = número de canales prestables en cada célula
- p = probabilidad de que un nuevo usuario participe en un balance de carga
- $salgodecac$ = Boolean que indica si una llamada que se inicia puede adquirir el canal propio de otra llamada en curso situada en una zona de load balancing de la célula.

Cada célula tiene NUMCELL células vecinas que es un parámetro cuyo valor puede decidir el usuario. La elección del vecino donde se produce una llamada en la zona de load balancing y de la probabilidad de estar en una zona de load balancing depende de si la célula considerada es la célula central o una célula externa.

- si ($i==0$)
 - $ve = (int) (ranf()*(NUMCELL-1))+1$
 - si ($ranf() \leq (NUMCELL-1)*p$)
 - zona lb = YES
 - si no
 - zona lb = NO
- si no
 - $ve = 0$
 - si ($ranf() \leq p$)
 - zona lb = YES
 - si no
 - zona lb = NO

Usamos una variable aux que contabilice el número de canales que la célula actual i presta a sus vecinas. En este caso, también depende de si la célula actual es la célula central o una célula externa.

- si (i==0) entonces
 - aux=jjs[i][ve]
- si no
 - para j \leftarrow 1 hasta NUMCELL-1 hacer
 - aux = aux + jjs[j][0]

El control de acceso de canal para una célula A viene determinado según el número de llamadas en curso dentro de la propia célula y de cada vecina. Tiene similitudes respecto a un sistema de dos células.

- si (0 < bt[i] < C) && (bttotal < NUMCELL*C) entonces
 - canalpropio=YES
 - is[i]++
 - isp[i]++
 - si llamada en zona de lb entonces
 - js[i]++
 - jsz[i][ve]++
- si (bt[i]== C) && (bttotal < NUMCELL*C) entonces
 - si (llamada en zona de lb) && (aux < B) && (bt[ve] < C) entonces
 - is[i]++
 - js[i]++
 - jsz[i][ve]++
 - jjs[i][ve]++
 - si (llamada en zona de lb) && (directo) && ((aux>=B) || (bt[ve]>=C))
 - la llamada se pierde: token_kill
 - si (llamada en zona de lb) && (directo=NO) && ((aux>=B) || (bt[ve]>=C))
 - si (i==0)
 - si hay canales propios en zona de lb de vebuscado entonces
 - is[i]++
 - isp[i]++
 - js[i]++
 - jsz[i][ve]++
 - salgodecac=YES
 - canalpropio=YES
 - si no
 - la llamada se pierde: token_kill
 - si no
 - la llamada se pierde: token_kill
 - si (llamada fuera de la zona de lb) && (directo=YES)
 - la llamada se pierde: token_kill

- si (llamada fuera de la zona de lb) && (directo=NO)
 - si hay canales propios en zona de lb de vebuscado entonces
 - is[i]++
 - isp[i]++
 - salgodecac=YES
 - canalpropio=YES
 - si no
 - la llamada se pierde: token_kill
- si no
 - la llamada se pierde: token_kill

El pseudocódigo de la función simular es parecido al de dos células, aunque añadimos la gestión del método indirecto para un sistema con más de una zona de load balancing, lo que incrementa su complejidad.

En caso del evento nuevo servicio debemos considerar la posibilidad de que una llamada que se inicia pueda tomar un canal propio de otra llamada en curso situada en una zona de load balancing de la célula, porque la primera llamada se encuentra situada en una zona donde no puede obtener un canal propio o prestado que no ha sido asignado. La llamada cuyo canal propio utiliza la nueva llamada, pasa a utilizar un canal prestado de la célula vecina. El canal propio liberado será utilizado por la nueva llamada.

- caso evento nuevo servicio
 - si salgodecac
 - si canalpropio
 - la llamada se pierde: token_kill
 - isp[i]--
 - jjs[i][vebuscado]++
 - canal_propio [tkill]=NO

La gestión de la finalización de las llamadas en un *cluster* de 3 células también implica cierto código que tenga en cuenta cuál de las zonas de load balancing es la involucrada. Es importante saber cuántas llamadas hay en las zonas de load balancing y cuáles de ellas utilizan canales propios de la célula y cuáles utilizan canales prestados.

- caso evento fin llamada
 - si (tk_time_lf < 0) entonces
 - si zona lb
 - si ((isp[i]==0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0))
 - is[i]--
 - isp[i]--
 - js[i]--

- jsz[i][ve]--
- si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - si (jsz[i][ve]==jjs[i][ve])
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si (jsz[i][ve]>jjs[i][ve])
 - si canalpropio=YES
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - si no
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
- si no
 - si ((isp[i]>0) & (jsz[i][ve]==0) & (jjs[i][ve]==0))
 - is[i]--
 - isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0) & (isp[i]>jsz[i][ve]))
 - is[i]--
 - isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0) & (is[i]>jsz[i][ve]))
 - is[i]--
 - isp[i]--

En la descripción del método de simulación de un sistema multicelular, el número total de células en un escenario de 3 anillos es de 37 células.

- NUMCELLTOTAL=37

En el escenario de 3 anillos debemos detallar para cada una de las células cuál es su vecina. Cada célula tendrá como máximo 6 vecinas.

- Para $0 \leftarrow i$ hasta (NUMCELLTOTAL -1) hacer
 - vecino[i][0]
 - vecino[i][1]
 - vecino[i][2]
 - vecino[i][3]
 - vecino[i][4]
 - vecino[i][5]

El proceso de simulación de un *cluster* es el mismo que se realiza para cada célula y sus vecinas. No todos los *clusters* son de 7 células si no que los *clusters* del tercer anillo están compuestos por 4 o por 3 células.

- NUMCELLCLUS=7
- NUMCELLCLU=1
- Para $0 \leftarrow i$ hasta (NUMCELLCLUS -1) hacer
 - si existe vecino[cell][i]
 - si $(1/arrv_m[vecino[cell][i]]!=0)$
 - NUMCELLCLU=NUMCELLCLU+1

La célula vecina donde se inicia una llamada en zona de load balancing o bien la célula a la que se dirige la llamada cuando hay movilidad se elige de forma aleatoria entre el número de células que componen el *cluster*.

- $ve=vecino[i][(int)(ranf()*(NUMCELLCLU-1))]$

La simulación que realizamos en un *cluster* se amplía a todas los *clusters* del sistema. El proceso de simulación no será entre *clusters* independientes puesto que todas las células están numeradas. Todos los cambios que se produzcan en una célula debido al procesamiento de una llamada en un *cluster*, se tendrán en cuenta para el procesamiento de otra llamada cuya célula vecina sea esa misma célula modificada en el otro *cluster*.

- Para $0 \leftarrow i$ hasta (NUMCELLTOTAL-1) hacer
 - creación de una nueva sesión: $tk=token_new()$
 - planificación de nueva llamada: $schedule(nueva\ llamada,0.0)$

El número de canales utilizados en la célula *i* tiene en cuenta el número de llamadas prestadas a las células vecinas y el número de llamadas que usan canales propios.

- Si (NUMCELLTOTAL==37)
 - Para $0 \leftarrow i$ hasta (NUMCELLTOTAL -1) hacer
 - Para $0 \leftarrow j$ hasta (NUMCELLCLUS -1) hacer
 - si existe vecino[i][j]
 - $bt[i]=bt[i]+jjs[vecino[i][j]][i]$
 - $bt[i]=bt[i]+isp[i]$

Para más información ver pseudocódigo en el capítulo 7.

4.3. Movilidad en un sistema multicelular.

El estudio de la movilidad en un sistema multicelular cualquiera se basa en una serie de variables aleatorias que determinan el tiempo de estancia de una llamada en una u otra célula. En la figura 48 podemos ver de forma gráfica un ejemplo de movilidad y las variables aleatorias de tiempo que se utilizan para gestionar la llamada.

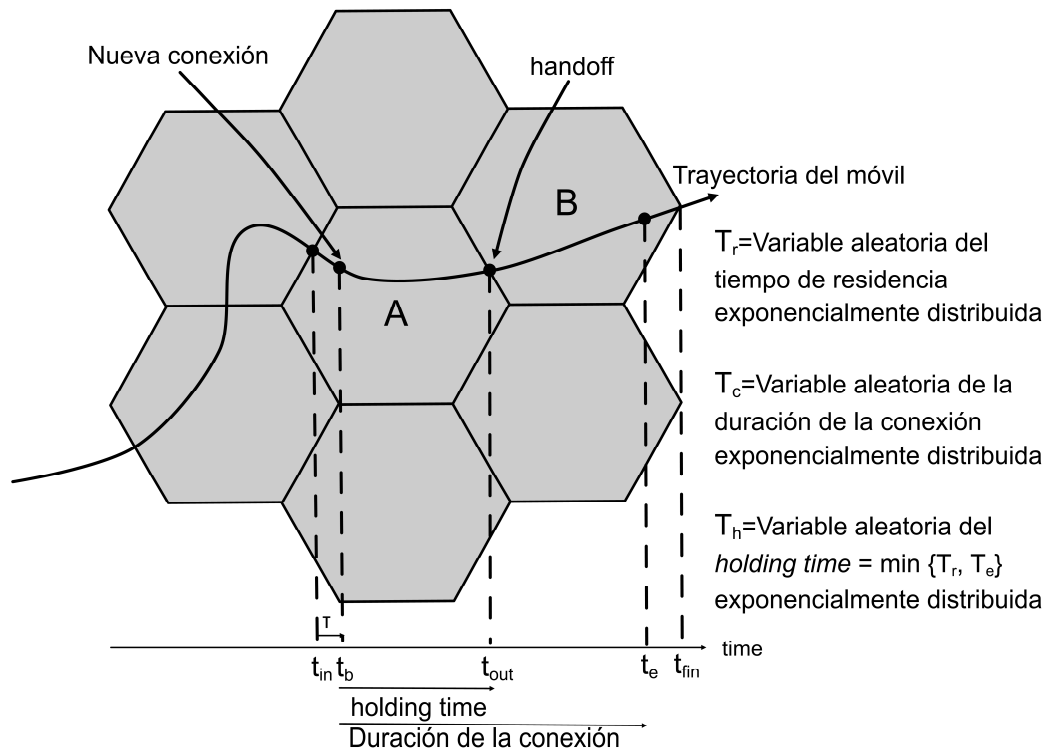


Fig. 48. Ejemplo de movilidad en un sistema multicelular.

Como vemos en la figura, se han utilizado las variables aleatorias siguientes:

- T_r =Variable aleatoria del tiempo de residencia
- T_c =Variable aleatoria de la duración de la conexión
- T_h =Variable aleatoria del holding time = $\min\{T_r, T_c\}$

Y los siguientes instantes de tiempo de las variables aleatorias:

- t_{in} : Instante de tiempo en el que entra el móvil en la célula A
- t_{out} : Instante de tiempo en el que el móvil pasa de la célula A a la célula B (*handoff* o *handover*)
- t_{fin} : Instante de tiempo en el que el móvil sale de la célula B
- t_b : Instante de tiempo en el que la llamada se inicia
- t_e : Instante de tiempo en el que la llamada termina

En la célula A, la duración de la conexión T_{cA} , es el tiempo desde que la llamada se inicia hasta que la llamada termina, es decir, desde t_b hasta t_e . El tiempo de residencia en la célula A, T_{rA} , es el tiempo desde que la llamada se inicia hasta que se produce el

handoff a la célula B, es decir, desde t_b hasta t_{out} . Por tanto, el holding time de la célula A es $T_{hA} = \min\{T_{rA}, T_{cA}\} = T_{rA}$.

En la célula B, la duración de la conexión T_{cB} , es el tiempo desde el *handoff* hasta que la llamada termina, es decir, desde t_{out} hasta t_e . El tiempo de residencia en la célula B, T_{rB} , es el tiempo desde el *handoff* hasta que el móvil sale de la célula B, es decir, desde t_{out} a t_{fin} . Por tanto, el holding time de la célula B es $T_{hB} = \min\{T_{rB}, T_{cB}\} = T_{cB}$.

En un sistema con movilidad, cada célula dispone de C canales que se pueden asignar a las llamadas, las cuales pueden ser llamadas nuevas o *handovers* que pasan de una célula a otra.

La movilidad en un sistema multicelular para las técnicas de balance de carga se ha gestionado considerando que la llamada se desplaza entre las distintas zonas que componen nuestro sistema multicelular. El desplazamiento entre células se produce cuando la llamada se mueve de la zona de load balancing de una célula a la zona fuera de load balancing de una célula vecina. No habrá desplazamiento entre células, aunque sí dentro de una célula cuando el usuario se mueve de zona fuera de load balancing a zona de load balancing.

Las variables que se utilizan en el estudio de la movilidad genérico para un sistema multicelular, se utilizarán en cada una de las zonas de load balancing o de no load balancing del modelo de simulación de las técnicas de balance de carga. El modelo genérico del la figura 48, tiene en cuenta variables aleatorias temporales entre células. En nuestro modelo de simulación las variables aleatorias temporales se contabilizan en cada una de las zonas de load balancing y zonas fuera de load balancing de nuestro sistema. En la figura 49, se observa el desplazamiento en línea recta de un móvil entre las zonas de load balancing y las zonas fuera de load balancing de un modelo multicelular. El modelo de la derecha es un modelo simplificado del sistema multicelular. El modelo simplificado es un modelo de 3 células que representa un sistema multicelular. Se ha realizado así por cuestiones de espacio.

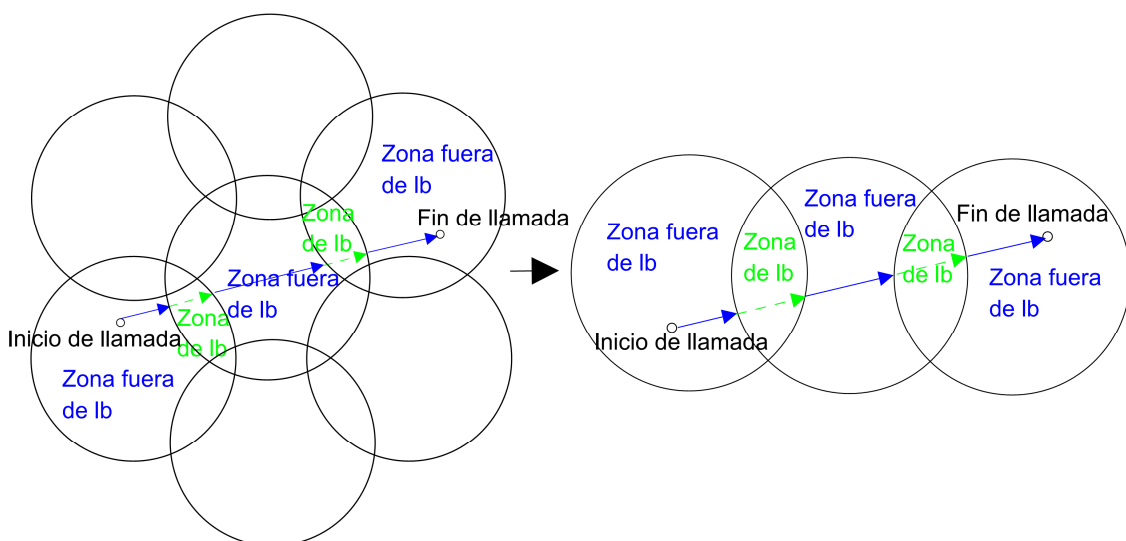


Fig. 49. Ejemplo de movilidad en un sistema multicelular compuesto por zonas de load balancing.

El procedimiento cuando se inicia el servicio de una llamada y el usuario es situado fuera de la zona de load balancing es el mismo que el sistema de dos células; la planificación del fin de la llamada depende del tiempo restante de la llamada, es decir, restamos la duración de la llamada menos la duración de residencia fuera de la zona de load balancing. Si este tiempo es mayor que cero entonces se planifica el fin de la llamada para una duración igual a la duración de residencia fuera de la zona de load balancing. En el caso de que sea menor o igual a 0, no existe dicha limitación, y se planifica para la duración de la llamada (ver figura 50).

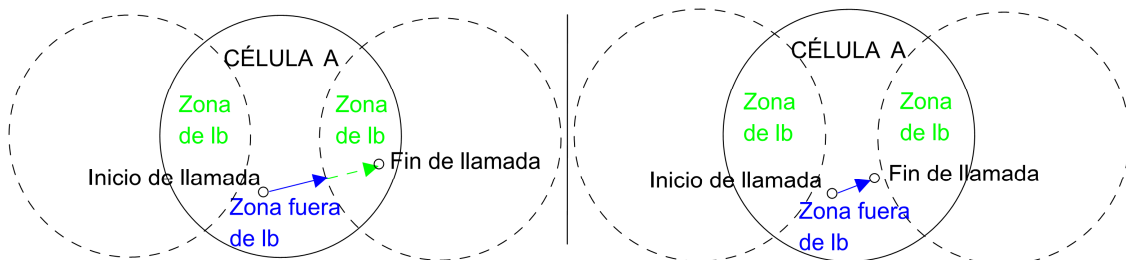


Fig. 50. Ejemplo de movilidad en un sistema multicelular cuando el usuario está situado fuera de la zona de load balancing.

Cuando se inicia el servicio de una llamada, el usuario está situado en zona de load balancing y la duración de la llamada es mayor que la duración de residencia en la zona de load balancing, el procedimiento es algo distinto al utilizado en el sistema de dos células, pues en este caso es importante saber cuál es la zona de no load balancing hacia la que se dirige la llamada, es decir, hacia qué célula vecina se dirige la llamada (ver figura 51). La manera que hemos considerado para determinar hacia qué célula se dirige la llamada cuando hay movilidad es mediante una probabilidad equitativa entre el número de células vecinas. La planificación del fin de llamada se hace para una duración igual a la duración de residencia en la zona de load balancing de la célula vecina elegida.

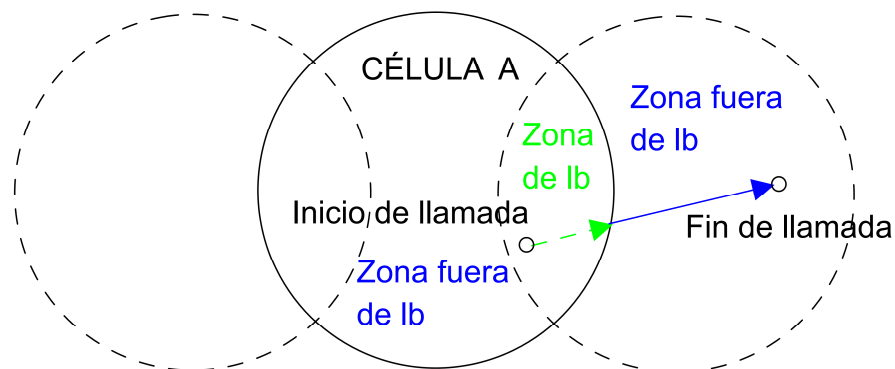


Fig. 51. Ejemplo de movilidad en un sistema multicelular cuando el usuario está situado en la zona de load balancing.

Para determinar cuándo finaliza una llamada, tenemos en cuenta el tiempo restante de la llamada tras atravesar una zona de load balancing (o de no load balancing) al igual que en el sistema de dos células. La planificación del fin de la llamada se realiza igual que en el sistema de dos células. Es importante saber hacia qué célula vecina se dirige la llamada cuando pasa de una zona de load balancing a una zona fuera de load balancing de una célula vecina para hacer una planificación correcta.

Si el tiempo restante de la llamada es mayor que cero y el usuario estaba situado en la zona de load balancing, la planificación del fin de la llamada depende del tiempo que le queda a la llamada, es decir, restamos el tiempo restante de la llamada menos la duración de residencia fuera de la zona de load balancing. Si este tiempo es mayor que cero, entonces se planifica el fin de la llamada para una duración igual a la duración de residencia fuera de la zona de load balancing. En el caso de que sea menor o igual a cero, no existe dicha limitación, y se planifica para la duración restante de la llamada.

Si el tiempo restante de la llamada es mayor que cero y el usuario estaba situado en la zona fuera de load balancing, la planificación del fin de la llamada depende del tiempo que le queda a la llamada, es decir, restamos el tiempo restante de la llamada menos la duración de residencia de la zona de load balancing. Si este tiempo es mayor que cero, entonces se planifica el fin de la llamada para una duración igual a la duración de residencia de la zona de load balancing. En el caso de que sea menor o igual a cero, no existe dicha limitación, y se planifica para la duración restante de la llamada.

En el estudio de la movilidad del sistema de dos células, la llamada se mueve entre zonas de una misma célula, no hay cambio del servicio de la llamada entre células. En el sistema celular, el cambio de servicio de la llamada entre células se produce en el paso de la zona de no load balancing de una célula a la zona de load balancing de otra célula si se cumplen unas condiciones de tráfico. En el desplazamiento de la llamada desde la zona fuera de load balancing hacia la zona de load balancing tampoco hay cambio de servicio entre células del sistema porque se considera que ambas zonas forman parte de la misma célula.

Cuando existe movilidad, la probabilidad de pérdidas se reduce en un sistema multicelular, como veremos en la representación gráfica.

4.4. Resultados y simulación gráfica en un cluster.

Siguiendo el mismo procedimiento que para el sistema de dos células, hemos realizado un estudio de un *cluster* mediante un programa de simulación tomando valores para los parámetros principales en cada tipo de técnica de balance.

La razón de hacer un estudio de un *cluster* es que el sistema multicelular lo implementamos como un conjunto de *clusters* relacionados en el programa de simulación. Por tanto, para asegurar que nuestro programa de simulación de un sistema multicelular funciona, antes debemos asegurar que el programa de simulación de un *cluster* es preciso. La forma de contabilizar la exactitud de los resultados de nuestro programa de simulación se hace a partir de un modelo analítico. El modelo analítico se ha estudiado mediante unos programas en Matlab. A continuación comparamos de forma gráfica los resultados obtenidos para un *cluster* de 3 células.

En la siguiente tabla se detallan los parámetros usados en nuestro programa de simulación para el caso de un *cluster* de 3 células. Básicamente son los mismos que para el caso de dos células.

<i>Posición del vector</i>	<i>Parámetros</i>	<i>Descripción</i>
1	C	Número de canales de cada célula
2	B	Número de canales prestables
3	p	Fracción de superficie de 2 células en la que se puede hacer <i>load balancing</i>
4	Tiempo de simulación	0 a infinito
5	Número de simulaciones	Simulaciones por línea de parámetros: 1-15
6	Porcentaje de transitorio	0% - 100%
7	MOBTY	0: NO, 1: YES
8	DIRECT	1: Método directo, 0: Método indirecto
9	NUMCELL	Número de células
10	μ_i	Tasa de servicio en la célula i, $E[s_i] = 1/\mu_i$
11	$\sigma(s_i)$	Desviación típica de s en la célula i
12	Distribución de s_i	1=exponencial, 2=erlang, ó 3=hyper-exponencial
13	$\lambda_{CENTRAL}$	Tasa de llegada en la célula central, $E[\tau_{CENTRAL}] = 1/\lambda_{CENTRAL}$
14	$\sigma(\tau_{CENTRAL})$	Desviación típica de $\tau_{CENTRAL}$
15	Distribución de $\tau_{CENTRAL}$	1=exponencial, 2=erlang, ó 3=hyper-exponencial
16	$\lambda_{VECINOS}$	Tasa de llegada en las células externas, $E[\tau_{VECINOS}] = 1/\lambda_{VECINOS}$
17	$\sigma(\tau_{VECINOS})$	Desviación típica de $\tau_{VECINOS}$
18	Distribución de $\tau_{VECINOS}$	1=exponencial, 2=erlang, ó 3=hyper-exponencial

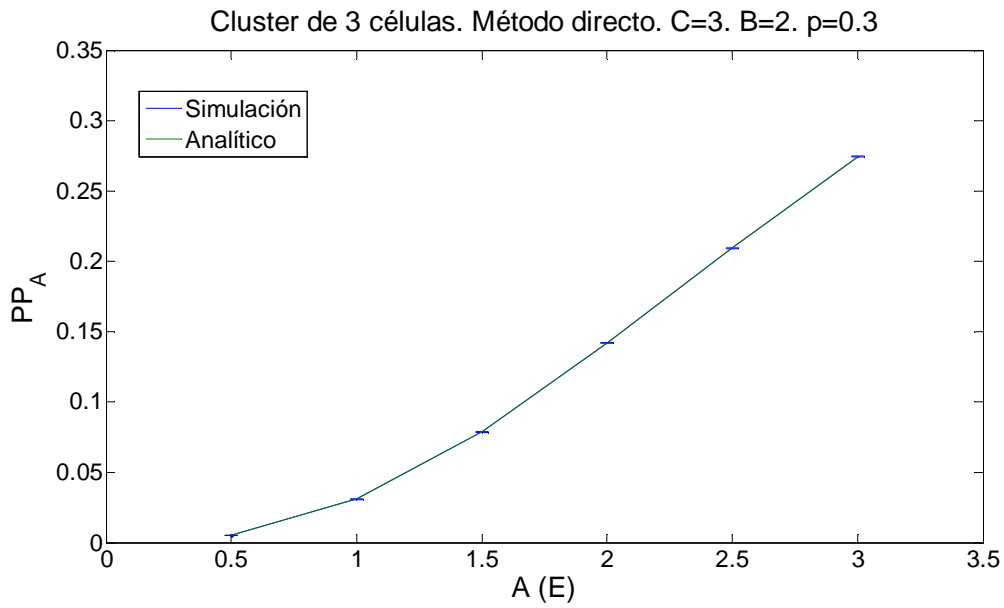
Tomamos unos valores ejemplo para los parámetros de nuestro sistema para el método directo.

<i>Parámetros de nuestro ejemplo</i>	
C	3
B	2
p	0.3
Tiempo de simulación	500000
Número de simulaciones	10
Porcentaje de transitorio	10%
MOBTY	0: NO
DIRECT	1: Método directo
NUMCELL	3
μ_i	1
$\sigma (s_i)$	1
Distribución de s_i	Exponencial
$\lambda_{CENTRAL}$	-
$\sigma (\tau_{CENTRAL})$	1
Distribución de $\tau_{CENTRAL}$	Exponencial
$\lambda_{VECINOS}$	-
$\sigma (\tau_{VECINOS})$	1
Distribución de $\tau_{VECINOS}$	Exponencial

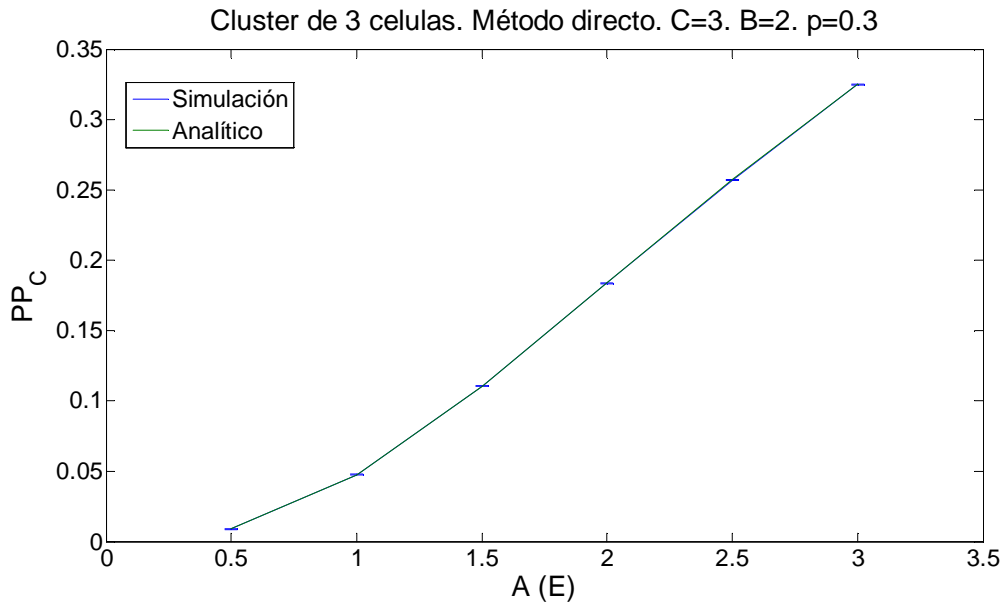
Tomamos 10 muestras de los resultados obtenidos con nuestro programa de simulación, es decir, 10 simulaciones por punto, y calculamos su media. Esta media será el valor que utilizemos para cada punto de nuestra gráfica. Además para cada punto representamos los intervalos de confianza del 95%, es decir, los intervalos donde la media real se encuentra con una probabilidad del 95%. Los intervalos de confianza se representan en las gráficas mediante dos pequeñas líneas azules horizontales unidas mediante una línea vertical. A menudo las líneas horizontales se encuentran muy próximas indicando que el intervalo de confianza es pequeño.

Representamos los valores obtenidos tanto en el método analítico como en la simulación y observamos la gran precisión de nuestro programa de simulación, puesto que efectivamente los puntos de nuestro modelo analítico se encuentran dentro de los intervalos de confianza.

Las gráficas 10 y 11 representan la evolución de la probabilidad de pérdidas en las células A y C para el método directo en un *cluster* de 3 células, A, B y C, en el que hacemos variar el tráfico ofrecido en las mismas. En la gráfica 10 y 11, observamos un comportamiento muy aproximado al método analítico de tres células. La probabilidad de pérdidas en la célula A y C para el método directo aumenta con el tráfico ofrecido en cada una de las tres células de forma creciente y con una pendiente cada vez mayor antes de la saturación.



Gráf. 10. PP_A / A para el método directo sin movilidad y mismo A en un *cluster* de 3 células, para C=3, B=2 y p=0.3.



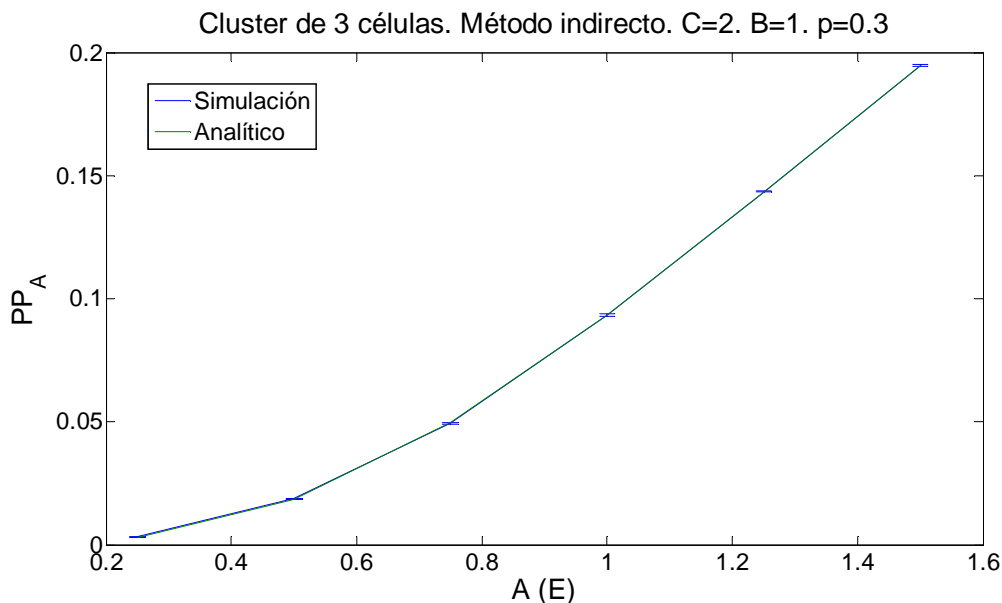
Gráf. 11. PP_C / A para el método directo sin movilidad y mismo A en un *cluster* de 3 células para C=3, B=2 y p=0.3.

A continuación vamos a proceder a evaluar la precisión de los resultados obtenidos en nuestro programa de simulación respecto a los resultados analíticos para el método indirecto.

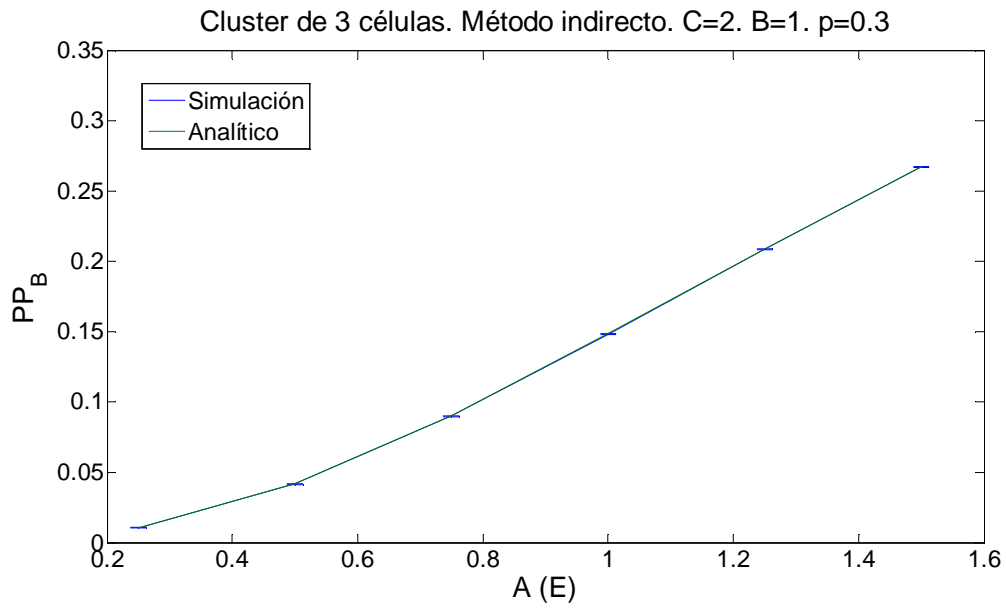
<i>Parámetros de nuestro ejemplo</i>	
C	2
B	1
p	0.3
Tiempo de simulación	500000
Número de simulaciones	10
Porcentaje de transitorio	10%
MOBTY	0: NO
DIRECT	0: Método indirecto
NUMCELL	3
μ_i	1
$\sigma (s_i)$	1
Distribución de s_i	Exponencial
$\lambda_{CENTRAL}$	-
$\sigma (\tau_{CENTRAL})$	1
Distribución de $\tau_{CENTRAL}$	Exponencial
$\lambda_{VECINOS}$	-
$\sigma (\tau_{VECINOS})$	1
Distribución de $\tau_{VECINOS}$	Exponencial

Una vez más observamos que nuestros resultados de la simulación son muy aproximados a los analíticos.

Las gráficas 12 y 13 representan la evolución de la probabilidad de pérdidas en las células A y B para el método indirecto en un *cluster* de 3 células, A, B y C, en el que hacemos variar el tráfico ofrecido en las mismas. En la gráfica 12 y 13, vemos que la probabilidad de pérdidas en la célula A y B para el método indirecto aumenta con el tráfico ofrecido en cada una de las tres células de forma creciente y con una pendiente cada vez mayor antes de la saturación, tal y como ocurría en el método directo.



Gráf. 12. PP_A / A para el método indirecto sin movilidad y mismo A en un *cluster* de 3 células, para C=2, B=1 y p=0.3.



Gráf. 13. PP_B / A para el método indirecto sin movilidad y mismo A en un *cluster* de 3 células para C=2, B=1 y p=0.3.

4.5. Resultados y simulación gráfica en un sistema multicelular.

A continuación hemos hecho un estudio de un sistema multicelular mediante un programa de simulación variando los parámetros principales para cada tipo de técnica de balance.

La forma de contabilizar que los resultados de nuestro programa de simulación son correctos se ha realizado teniendo en cuenta que del sistema de 3 anillos, únicamente en 3 células se originan llamadas y se prestan canales, para poder comparar los resultados con los del método analítico de 3 células. A continuación compararemos de forma gráfica los resultados obtenidos para el sistema analítico y el sistema de simulación. El modelo analítico se ha estudiado mediante unos programas en Matlab. El fin de este procedimiento es asegurar que nuestro programa de simulación funciona.

En la siguiente tabla se detallan los parámetros usados en nuestro programa de simulación para el caso de un sistema multicelular compuesto por 3 anillos, en total 37 células. Básicamente son los mismos que para el caso de dos células.

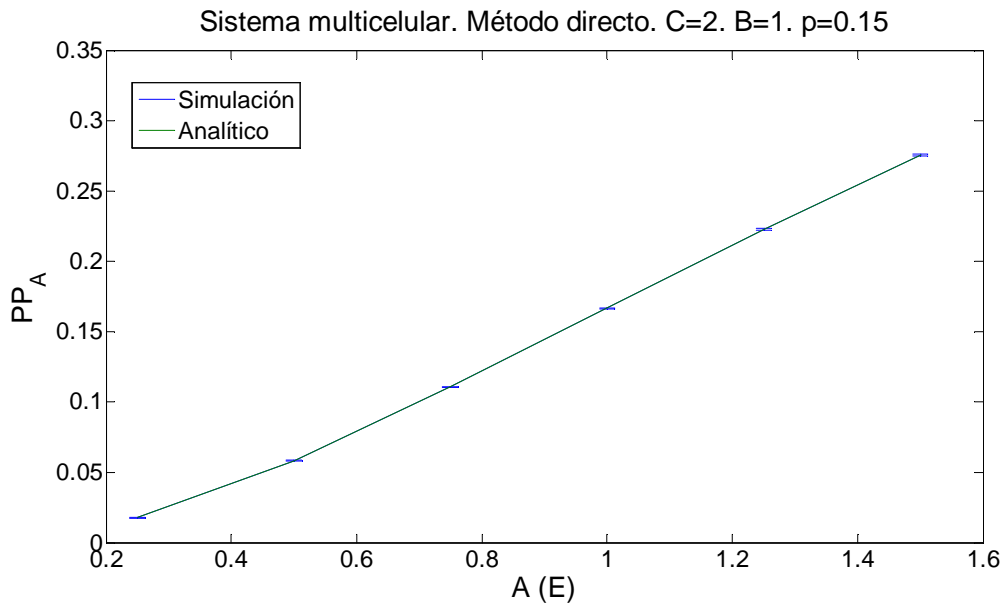
<i>Posición del vector</i>	<i>Parámetros</i>	<i>Descripción</i>
1	C	Número de canales de cada célula
2	B	Número de canales prestables
3	p	Fracción de superficie de 2 células en la que se puede hacer <i>load balancing</i>
4	Tiempo de simulación	0 a infinito
5	Número de simulaciones	Simulaciones por línea de parámetros: 1-15
6	Porcentaje de transitorio	0% - 100%
7	MOBTY	0: NO, 1: YES
8	DIRECT	1: Método directo, 0: Método indirecto
9	$\sigma(\tau_i)$	Desviación típica de τ en la célula i
10	Distribución de τ_i	1=exponencial, 2= rlang, ó 3=hyper-exponencial
11	$\sigma(s_i)$	Desviación típica de s en la célula i
12	Distribución de s_i	1=exponencial, 2=erlang, ó 3=hyper-exponencial
13 a 13+36	λ_i	Tasa de llegada en la célula i, $E[\tau_i] = 1/\lambda_i$
50 a 50+36	μ_i	Tasa de servicio en la célula i, $E[s_i] = 1/\mu_i$

Para ello hemos tomado unos valores ejemplo para los parámetros de nuestro sistema. A continuación se muestran los valores que hemos mantenido fijos en nuestro ejemplo.

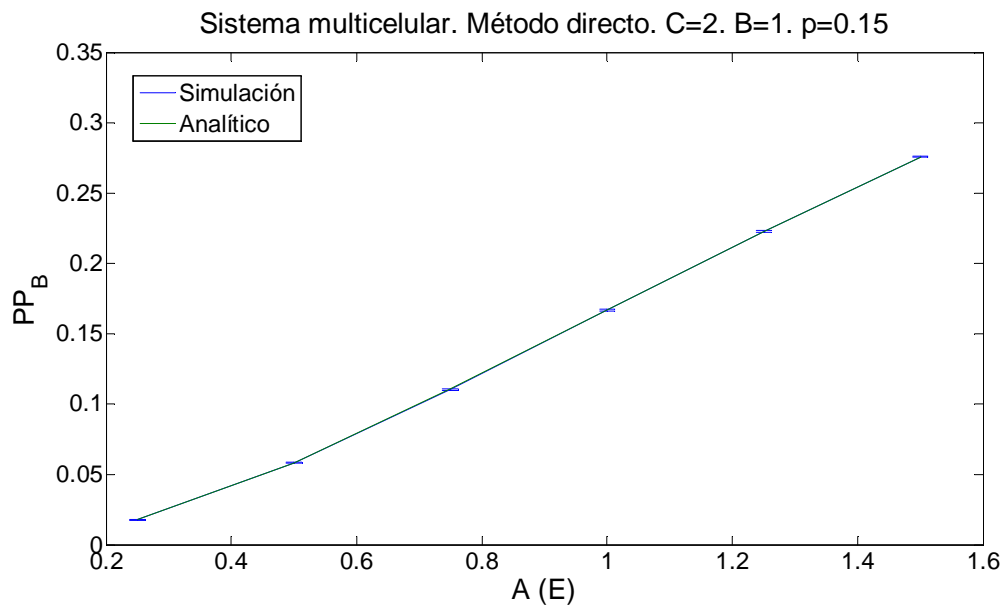
<i>Parámetros de nuestro ejemplo</i>		
C	2	1
B		1
p		0.15
Tiempo de simulación		500000
Número de simulaciones		10
Porcentaje de transitorio		10%
MOBTY		0: NO
DIRECT	1: Método directo	0: Método indirecto
$\sigma(\tau_i)$		1
Distribución de τ_i		Exponencial
$\sigma(s_i)$		1
Distribución de s_i		Exponencial
λ_1		-
λ_2		-
λ_3		-
λ_{resto}		0
μ_1		1
μ_2		1
μ_3		1
μ_{resto}		0

Tomamos 10 muestras de los resultados obtenidos con nuestro programa de simulación, es decir, 10 simulaciones por punto, y calculamos su media. Esta media será el valor que utilizemos para cada punto de nuestra gráfica. Además para cada punto representamos los intervalos de confianza del 95% mediante dos pequeñas líneas azules paralelas.

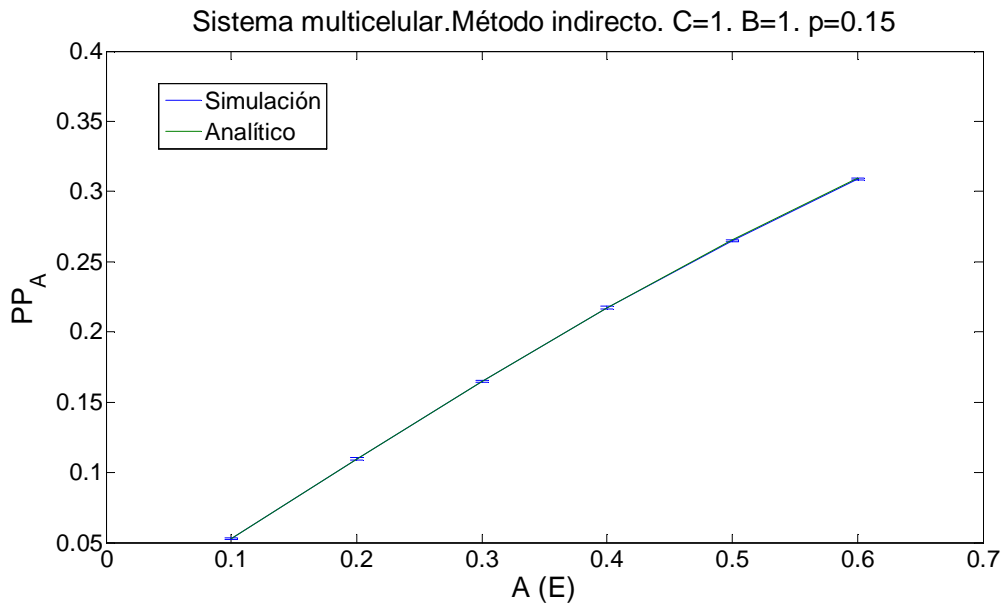
Representamos los valores obtenidos tanto en el método analítico como en la simulación y observamos la considerable precisión de nuestro programa de simulación, puesto que los puntos de nuestro modelo analítico se encuentran dentro de los intervalos de confianza tanto en el método directo como en el indirecto. Para 10 simulaciones por punto y un tiempo de simulación de 500.000 unidades, los errores cometidos en los métodos directo e indirecto han sido del orden de 10^{-3} para probabilidades de bloqueo $P_b < 5\%$ y del orden del 10^{-4} y 10^{-5} para $P_b > 5\%$.



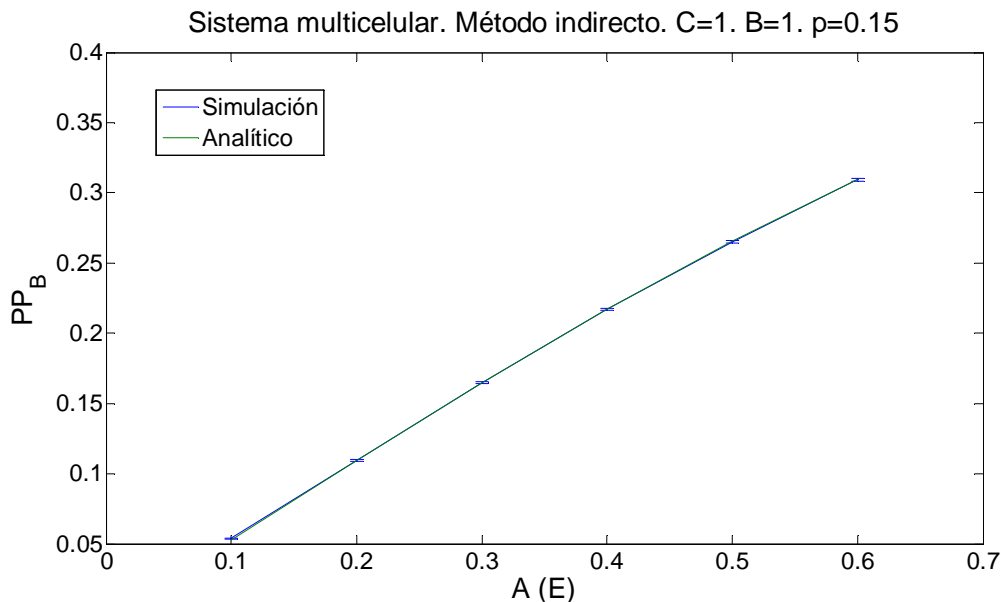
Gráf. 14. PP_A / A para el método directo sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para C=2, B=1 y p=0.15.



Gráf. 15. PP_B / A para el método directo sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para C=2, B=1 y p=0.15.



Gráf. 16. PP_A / A para el método indirecto sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=1$, $B=1$ y $p=0.15$.



Gráf. 17. PP_B / A para el método indirecto sin movilidad y mismo A en un sistema multicelular de tres anillos con 3 células operativas para $C=1$, $B=1$ y $p=0.15$.

Vemos que el comportamiento de las graficas 14 a 17 es muy parecido, aunque la pendiente de la probabilidad de pérdidas para el método indirecto con $C=1$ es mayor que la del método directo con $C=2$. Tenemos en cuenta que el intervalo del tráfico ofrecido A para el método directo (de 0.25 a 1.5) es mayor que el intervalo del método indirecto (de 0.1 a 0.6). Si tomamos los valores del método analítico, la pendiente de la curva PP_A / A para el método directo es 0.2064, mientras que en el método indirecto es 0.5128. Si hubiésemos representado el método directo con $C=1$, las pendientes de ambas gráficas serían casi iguales.

Nuestro último paso es simular un escenario real compuesto por 3 anillos y 37 células. Efectuamos tres simulaciones por punto y contamos con un número de canales más o menos realista, en total 10 canales. Las simulaciones que podemos realizar para

menor número de células utilizan mayor número de canales, pero para sistemas multicelulares el tiempo de simulación se incrementa mucho. Por ejemplo, para un sistema multicelular más o menos realista de 30 canales por célula y 10 simulaciones por punto, el tiempo que necesitamos para hacer una simulación es de 4 a 5 horas, lo cual no es factible para la realización de gráficas y optamos por un escenario menos exigente.

A continuación mostramos los parámetros utilizados para las representaciones gráficas:

<i>Parámetros de nuestro ejemplo</i>		
C		10
B		2
p		0.15
Tiempo de simulación		500000
Número de simulaciones		3
Porcentaje de transitorio		10%
MOBTY	0: NO	1: SÍ
DIRECT	1: Método directo	0: Método indirecto
$\sigma(\tau_i)$		1
Distribución de τ_i		Exponencial
$\sigma(s_i)$		1
Distribución de s_i		Exponencial
λ_i		-
μ_i		1

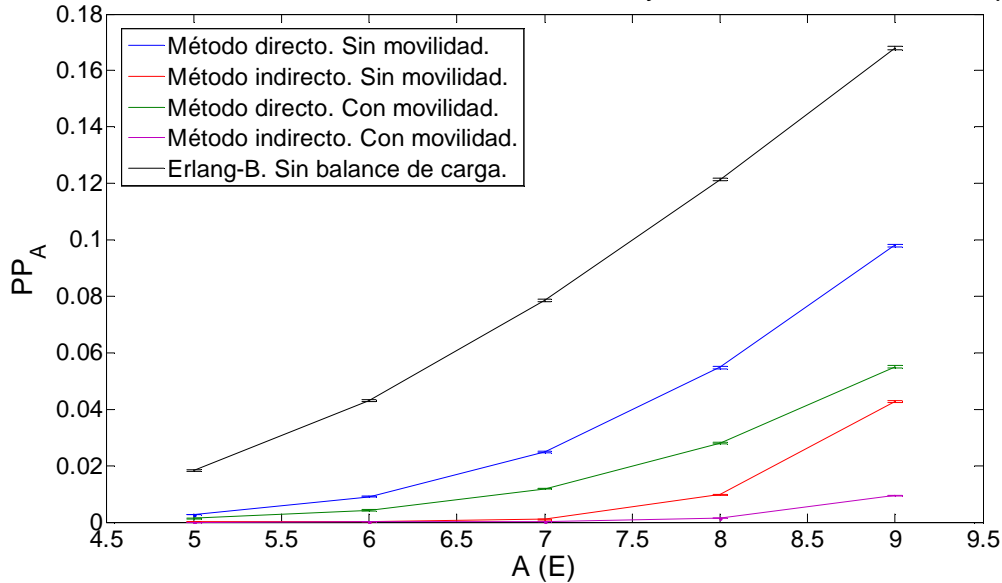
En la gráfica 18 hemos representado las probabilidades de pérdidas en la célula central para los distintos métodos de balance de carga dinámico estudiados (directo, indirecto, con y sin movilidad) y las hemos comparado con la probabilidad de pérdidas del sistema multicelular cuando no utiliza un balance de carga, para $C=10$, $B=2$ y $p=0.15$.

Observamos que la probabilidad de pérdidas para las técnicas de balance de carga es considerablemente menor que para las técnicas sin balance de carga. Para un escenario realista con probabilidades de pérdidas menores del 10%, la probabilidad de pérdidas es mayor para el método directo sin movilidad, luego para el método directo con movilidad, después para el indirecto sin movilidad y por último es menor para el método indirecto con movilidad.

Para probabilidades de pérdidas menores del 10%, con el método indirecto siempre conseguimos menor probabilidad de pérdidas que el método directo aunque para el método directo se asuma movilidad.

Para mayor tráfico ofrecido, el método directo con movilidad presenta menor probabilidad de pérdidas que el método indirecto sin movilidad. Para condiciones de alto tráfico de llamadas, el hecho de que todos los usuarios, incluidos los envueltos en una llamada, participen en el balance de carga es menos relevante que para bajo tráfico de llamadas. Sin embargo, el estudio de probabilidades de pérdidas de más del 10% no suele ser realista.

Sistema multicelular. Método directo, indirecto, con y sin movilidad. C=10. B=2. $\rho=0.15$

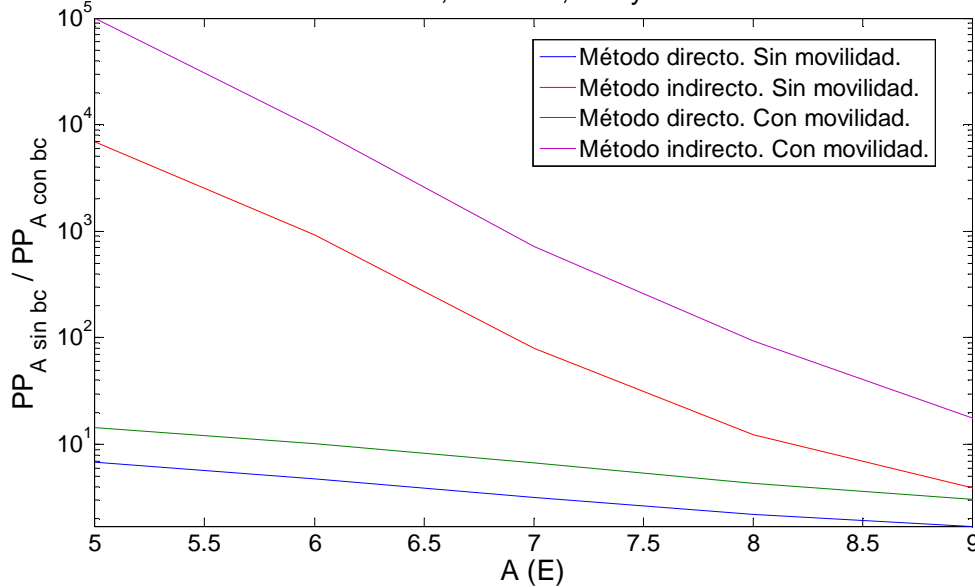


Gráf. 18. PP_A / A para los métodos directo, indirecto, con y sin movilidad en un sistema multicelular de tres anillos para C=10, B=2 y $\rho=0.15$.

En la gráfica 19 hemos representado el valor relativo de la probabilidad de pérdidas en la célula central sin método de balanceo de carga respecto a la probabilidad de pérdidas con los métodos de balanceo de carga, para el mismo caso anterior con C=10, B=2 y $\rho=0.15$.

Según la gráfica 18, para el caso Erlang-B, la probabilidad de pérdidas en la célula A es menor del 10% para un tráfico ofrecido menor de 7 Erlangs. Tomamos este valor de tráfico ofrecido como referencia en la gráfica 19. Observamos que para tráficos ofrecidos menores que 7 E, la probabilidad de pérdidas para las técnicas de balanceo de carga queda reducida casi en un orden de magnitud para el método directo con y sin movilidad y de 2 a 5 órdenes de magnitud para los métodos indirectos con y sin movilidad.

Sistema multicelular. Método directo, indirecto, con y sin movilidad. C=10. B=2. p=0.15.



Gráf. 19. $PP_{A \text{ sin bc}} / PP_{A \text{ con bc}}$ para los métodos directo, indirecto, con y sin movilidad en un sistema multicelular de tres anillos para C=10, B=2 y p=0.15.

Por último, mostramos unas figuras del sistema multicelular (figura 52 y 53) donde se representa, para los métodos directo sin movilidad e indirecto con movilidad, el error cometido en la probabilidad de pérdidas para un tráfico ofrecido de 7 E respecto a la probabilidad de pérdidas de la célula central en %, para C=10, B=2 y p=0.15. Observamos como en el tercer anillo obtenemos probabilidades de pérdidas que difieren de las de los anillos internos debido a que están constituidos por *clusters* más pequeños y la capacidad de balance dinámico de carga se reduce.

Para el método directo sin movilidad el error cometido es menor del 0.25% en el primer anillo y del 5% en el segundo. En cuanto al método indirecto con movilidad, el error es menor del 6% en el primer anillo y menor del 24% en el segundo. En el método indirecto con movilidad las variaciones son más grandes puesto que las probabilidades de pérdidas son muy pequeñas (del orden de 10^{-4}).

Error en la PP respecto a la célula central (%) para $T_0 = 7E$

simul/punto=3
 tsimul=500000
 Método directo, sin movilidad
 37 células (3 anillos)
 $T_0=7$
 $PPA=0,024827$

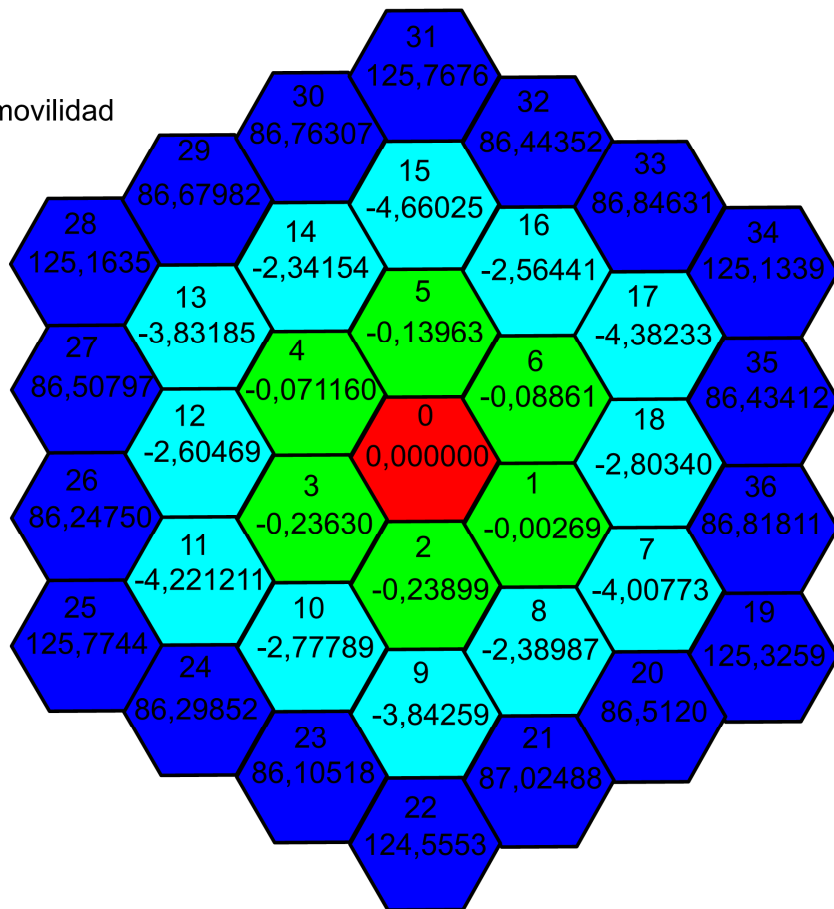


Fig. 52. Representación del error en la probabilidad de pérdidas respecto a la célula central para $T_0 = 7E$ en el sistema multicelular de tres anillos para el método directo sin movilidad y $C=10$, $B=2$ y $p=0.15$.

Error en la PP respecto a la célula central (%) para $T_0 = 7E$

simul/punto=3
 tsimul=500000
 Método indirecto, con movilidad
 37 células (3 anillos)
 $T_0=7$
 $PPA=0,000110$

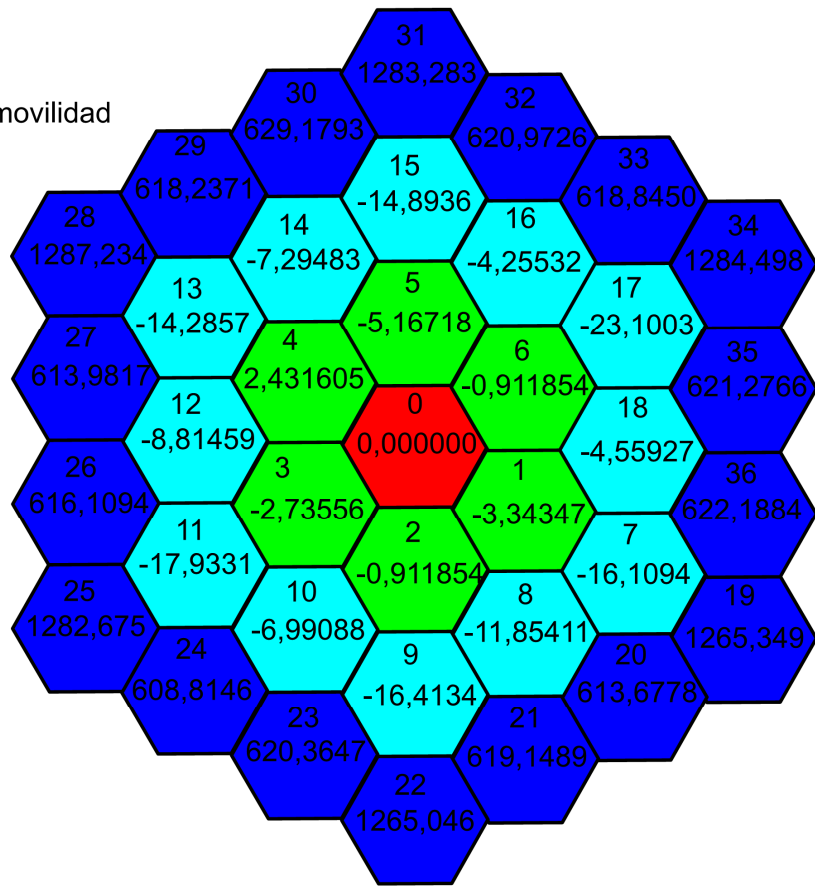


Fig. 53. Representación del error en la probabilidad de pérdidas respecto a la célula central para $T_0 = 7E$ en el sistema multicelular de tres anillos para el método indirecto con movilidad y $C=10$, $B=2$ y $p=0.15$.

5. Conclusiones.

Las principales conclusiones que obtenemos de nuestro trabajo son las siguientes:

- El balance de carga de las redes celulares es un proceso que ayuda al rendimiento y mejora las prestaciones de dichas redes. Es una técnica de gestión de tráfico que permite aliviar el problema de los picos de tráfico o *hot spots*.
- El balance de carga reduce la probabilidad de bloqueo y evita problemas como la congestión y la terminación forzosa de llamadas. Pero el problema del bloqueo no se soluciona únicamente haciendo un balance de carga. Es importante que la red celular esté bien dimensionada por el operador.
- Las técnicas de balance de carga dinámico tratan de resolver el problema de los *hot spots* de una forma dinámica, tal que el exceso de tráfico de estas células será servido por células adyacentes.
- Dichas técnicas las clasificamos en aquellas que usan el método directo y en las que usan el método indirecto. En las primeras, sólo las nuevas peticiones participan en el balance de carga cuando las células han utilizado todos sus canales locales o propios, y en las segundas, todos los usuarios, incluidos los que tienen llamadas en curso, participan en el balance de carga.
- Antes de la realización de este proyecto no teníamos ningún conocimiento de simulación de sistemas multicelulares con movilidad del grupo del director del proyecto. Los estudios realizados hasta ahora únicamente se basaban en modelos estáticos. No obstante, la movilidad de un usuario es una particularidad de las comunicaciones celulares y no puede ser obviada en el estudio de estas redes si queremos que nuestro análisis sea realista.
- Hemos realizado un estudio de las prestaciones de los sistemas de balance de carga que tenga en cuenta la movilidad de los usuarios y hemos hallado la probabilidad de bloqueo de un sistema multicelular con movilidad compuesto por 37 células para las técnicas de balance dinámico de carga.
- Los modelos analíticos de balance de carga realizados en programas de Matlab sirven para asegurar que nuestro programa de simulación realizado en C con librerías SMPL funciona. La solución analítica sólo es posible para sistemas pequeños. La solución para sistemas más complejos se obtiene mediante simulación.
- Una vez obtenidos los resultados de nuestra simulación y comparados con los resultados del modelo analítico, extraemos resoluciones acerca de las técnicas de balance dinámico estudiadas en los diversos tipos de sistemas.
- Hemos comparado los sistemas de balance de carga con sistemas de pérdidas Erlang-B. Los resultados obtenidos en las técnicas de balance dinámico

constatan que la probabilidad de bloqueo obtenida mediante este tipo de técnicas es menor que las técnicas no dinámicas, las cuales no prestan canales. Hemos visto que para un sistema realista la probabilidad de pérdidas para las técnicas de balance de carga queda reducida casi en un orden de magnitud para el método directo con y sin movilidad y de 2 a 5 órdenes de magnitud para los métodos indirectos con y sin movilidad.

- Para un escenario realista con probabilidades de pérdidas menores del 10%, la probabilidad de pérdidas es mayor para el método directo sin movilidad, luego para el método directo con movilidad, después para el indirecto sin movilidad y por último es menor para el método indirecto con movilidad.
- Los resultados obtenidos cuando existe movilidad son mejores que cuando no existe, es decir, para la misma intensidad de tráfico hay una menor probabilidad de bloqueo cuando hay movilidad tanto en el método directo como en el indirecto. El motivo principal es que la movilidad reduce el channel holding time, que es el tiempo de ocupación de los recursos y proporciona flexibilidad al sistema, puesto que los canales se liberan más rápidamente, aunque la media del channel holding time sea la misma que en sistemas sin movilidad.

6. Referencias.

- [1] Ozan K. Tonguz and Evsen Yanmaz, "The Mathematical Theory of Dynamic Load Balancing in Cellular Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, December 2008.
- [2] T.J. Kahwa and N.D. Georganas, "A Hybrid Channel Assignment Scheme in Large-Scale, Cellular-Structured Mobile Communication Systems," *IEEE Trans. Comm.*, vol. 26, no. 4, pp. 432-438, Apr. 1978.
- [3] X. Wu, B. Mukherjee, and S.-H.G. Chan, "MACA—An Efficient Channel Allocation Scheme in Cellular Networks," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '00)*, vol. 3, pp. 1385-1389, 2000.
- [4] B. Eklundh, "Channel Utilization and Blocking Probability in a Cellular Mobile Telephone System with Directed Retry," *IEEE Trans. Comm.*, vol. 34, no. 4, pp. 329-337, Apr. 1986.
- [5] H. Jiang and S.S. Rappaport, "CBWL: A New Channel Assignment and Sharing Method for Cellular Communication Systems," *IEEE Trans. Vehicular Technology*, vol. 43, no. 2, pp. 313-322, May 1994.
- [6] S.K. Das, S.K. Sen, and R. Jayaram, "A Dynamic Load Balancing Strategy for Channel Assignment Using Selective Borrowing in Cellular Mobile Environment," *Wireless Networks*, vol. 3, no. 5, pp. 333-347, Oct. 1997.
- [7] I. Katzela and M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey," *IEEE Personal Comm.*, vol. 3, no. 3, pp. 10-31, June 1996.
- [8] *Conmutadores de paquetes. Arquitectura y prestaciones.* Jorge Martínez y Vicente Casares.
- [9] *Simulating Computer Systems.* M. H. MacDougall. The Hit Press. 1987
- [10] R.B. Cooper, *Introduction to Queueing Theory.* The MacMillan, 1972.
- [11] L. Kleinrock, *Queueing Systems Volume I: Theory.* John Wiley & Sons, 1975.

7. Anexos.

7.1. Descripción del método de simulación: El entorno de simulación SMPL.

SMPL [9] es un conjunto de funciones de librería del lenguaje de programación C que proporcionan un lenguaje de simulación orientado a eventos. SMPL tiene tres tipos de entidades básicas:

- **Estaciones de servicio:** Son los recursos del sistema. En concreto se caracterizan por un conjunto de servidores y la correspondiente cola para alojar a los clientes que esperan ser atendidos. SMPL posee funciones para definir estaciones de servicio, reservarlas, abandonarlas, o interrogar acerca de su estado.
- **Marcas (tokens):** Son las unidades que modelan el comportamiento dinámico del sistema simulado. Pueden identificar clientes individuales del sistema o bien clases de estos.
- **Eventos:** Representan los cambios que se producen en el sistema. SMPL proporciona funciones para registrarlos y seleccionarlos de acuerdo a su tiempo de ocurrencia. Cada evento está identificado por un número, por el tiempo de simulación en el que ocurre y por el identificador de la marca involucrada.

Las funciones de SMPL empleadas para la simulación son las siguientes:

- **simpl (m, s);**
int m; char *s;
El parámetro m estará a '0' para desactivar el entorno interactivo de ejecución y s es un puntero a un conjunto de caracteres que describen el modelo. Esta función se encarga de inicializar las estructuras de datos, el reloj que registra el tiempo de simulación y el intervalo de medida.
- **schedule (ev, te, tkn);**
int ev, tkn; real te;
Esta función planifica un evento con número ev en la tabla de eventos. te es el tiempo incremental futuro a partir del instante actual en que ocurrirá el evento y tkn es la marca asociado a dicho evento. La función añade el tiempo que refleje el reloj de simulación a te y posiciona el evento en la tabla de eventos que está ordenada en valores ascendentes de tiempo de ocurrencia del evento. En algunos casos es interesante utilizar un te=0 para planificar el siguiente evento. El número ev es una marca que deja el programador para saber qué tipo de tareas deberá realizar para gestionar dicho evento cuando éste ocurra. Por ejemplo, ev podría indicar el evento salida de un cliente de una estación de servicio.
- **cause (ev, tkn);**
int *ev, *tkn;

Esta función extrae el siguiente evento de la cabecera de la tabla de eventos y avanza el reloj de simulación hasta el instante en el que se debe producir el evento. La función devuelve el número del evento y el número de la marca.

- **time();**
time() devuelve el tiempo actual de simulación. En la inicialización de SMPL se pone a '0' y es incrementado a través de la función cause() a medida que el programa de simulación inicia la planificación del siguiente evento.

7.2. Descripción del método de simulación en un sistema de dos células.

El pseudocódigo del control de acceso de canal en un sistema de dos células es el siguiente:

- i_A = llamadas en curso en célula A
- i_B = llamadas en curso en célula B
- j_A = llamadas en curso dentro de la zona de load balancing de célula A
- j_B = llamadas en curso dentro de la zona de load balancing de célula B
- C = número de canales en cada célula
- B = número de canales prestables en cada célula
- p = probabilidad de que un nuevo usuario participe en un balance de carga

- si $(0 < i_A < C) \ \&\& \ (i_B + i_A < 2C)$ entonces
 - i_{A++}
 - si llamada en zona de lb entonces
 - j_{A++}
- si $(C + B > i_A \geq C) \ \&\& \ (i_B + i_A < 2C)$ entonces
 - si llamada en zona de lb entonces
 - i_{A++}
 - j_{A++}
 - si no
 - si hay canales propios en zona de lb entonces
 - si directo
 - token_kill
 - si no
 - i_{A++}
 - si no
 - token_kill
- si $(i_A = C + B \ \&\& \ i_A + i_B < 2C) \ || \ (i_A + i_B = 2C)$ entonces
 - token_kill

El pseudocódigo de la función simular se expone a continuación:

- $serv_m[i], arrv_m[i]$ = medias de tasa de servicio y tasa de llamadas en célula i
- $serv_s[i], arrv_s[i]$ = desviación estándar de tasa de servicio y tasa de llamadas en célula i
- $serv_d[i], arrv_d[i]$ = tipo de distribución de tasa de servicio y tasa de llamadas en célula i
- sim_duratn = tiempo de simulación
- $cell_OC[i]$ = llamadas ofrecidas en célula i
- $cell_LC[i]$ = llamadas perdidas en célula i
- $cell_AC[i]$ = llamadas aceptadas en célula i
- $next_call_t$ = tiempo de la siguiente llamada
- $call_st$ = duración de la llamada
- $call_lbt,$ = duración de residencia en la zona de load balancing

- call_nolbt = duración de residencia fuera de la zona lb
- tk_time_lf = tiempo restante de llamada
- PP[i] = probabilidad de pérdidas en la célula i

- inicialización de tokens: token_init
- para $i \leftarrow 0$ hasta NUMCELL hacer
 - generación nuevo token: token_new
 - planificación evento nueva llamada: schedule(nueva llamada,0)
- planificación evento transitorio: schedule(evento transitorio, tiempo transitorio)
- mientras (tiempo de simulación > tiempo actual) hacer
 - extracción del evento de tabla de eventos: cause(event)
 - seleccionar event
 - caso evento nueva llamada
 - planificación evento nuevo servicio: schedule(nuevo servicio,0)
 - generación nuevo token
 - seleccionar arrv_d[i]
 - caso expon
 - next_call_t = expntl(arrv_m[i])
 - caso erlng y caso hyper del mismo modo
 - planificación evento nueva llamada: schedule(nueva llamada, next_call_t)
 - caso evento nuevo servicio
 - cell_OC[i]++
 - si control de acceso entonces
 - cell_AC[i]++
 - seleccionar serv_d[i]
 - caso expon
 - call_st=expntl(serv_m[i])
 - call_lbt=expntl(serv_m[i]/p)
 - call_nolbt=expntl(serv_m[i]/(1-p))
 - caso erlng y caso hyper del mismo modo
 - si llamada en zona de lb: tk_zone_lb[tk]
 - si no MOBTY entonces
 - call_lbt=call_st
 - $x = \text{call_st} - \text{call_lbt}$
 - si $(x \leq 0)$ entonces
 - planificación evento fin llamada: schedule(fin llamada,call_st)
 - tk_time_lf=0
 - si no
 - planificación evento fin llamada: schedule(fin llamada,call_lbt)
 - tk_time_lf=x
 - si no
 - si no MOBTY entonces
 - call_nolbt=call_st
 - $x = \text{call_st} - \text{call_nolbt}$

- si ($x \leq 0$) entonces
 - planificación evento fin llamada: `schedule(fin llamada, call_st)`
 - `tk_time_lf=0`
 - si no
 - planificación evento fin llamada: `schedule(fin llamada, call_nolbt)`
 - si no
 - `cell_LC[i]++`
 - caso evento fin llamada
 - seleccionar `serv_d[i]`
 - caso expon
 - `call_lbt=expntl(serv_m[i]/p)`
 - `call_nolbt=expntl(serv_m[i]/(1-p))`
 - caso erlng y caso hyper del mismo modo
 - si (`tk_time_lf > 0`) entonces
 - si llamada en zona de lb entonces
 - `js[i]--`
 - llamada fuera de zona de lb
 - `x=tk_time_lf - call_nolbt`
 - si ($x \leq 0$) entonces
 - `tk_time_lf=0`
 - `schedule(fin llamada, tk_time_lf)`
 - si no
 - `tk_time_lf=x`
 - `schedule(fin llamada, call_nolbt)`
 - si no
 - `js[i]++`
 - llamada en zona de lb
 - `x=tk_time_lf - call_lbt`
 - si ($x \leq 0$) entonces
 - `tk_time_lf=0`
 - `schedule(fin llamada, tk_time_lf)`
 - si no
 - `tk_time_lf=x`
 - `schedule(fin llamada, call_lbt)`
 - si no
 - `is[i]--`
 - si llamada en zona de lb
 - `js[i]--`
 - `token_kill`
 - caso evento transitorio
 - `cell_OC[i]=0`
 - `cell_LC[i]=0`
 - `cell_AC[i]=0`
- $PP[i]=(cell_LC[i])/(cell_OC[i])$

7.3. Descripción del método de simulación en un cluster.

El pseudocódigo del control de acceso de canal en un cluster es el siguiente:

- ve = célula vecina elegida
- is[i] = número de llamadas en curso en la célula i (ya sean canales propios o prestados de otras células)
- isp[i] = número de llamadas en curso que utilizan canales propios de la célula i
- js[i] = número de llamadas en curso en zona lb de la célula i (ya sean propios o prestados de otras células)
- jsz[i][ve] = número de llamadas en curso en zona lb de la célula i que comparte la zona de lb de la célula ve (ya sean propios o prestados de ve)
- jjs[i][ve] = número de canales que la célula i toma prestados de la célula ve
- bt[i] = número de canales utilizados en la célula i ya sean canales propios o canales que está prestando a otras células
- C = número de canales en cada célula
- B = número de canales prestables en cada célula
- p = probabilidad de que un nuevo usuario participe en un balance de carga
- NUMCELL = número de células que forman el cluster

- si (i==0)
 - ve = (int) (ranf()*(NUMCELL-1))+1
 - si (ranf() <= (NUMCELL-1)*p)
 - zona lb = YES
 - si no
 - zona lb = NO
- si no
 - ve = 0
 - si (ranf() <= p)
 - zona lb = YES
 - si no
 - zona lb = NO

- si (i==0) entonces
 - aux=jjs[i][ve]
- si no
 - para j ← 1 hasta NUMCELL-1 hacer
 - aux = aux + jjs[j][0]

- si (0 < bt[i] < C) && (bttotal < NUMCELL*C) entonces
 - canalpropio=YES
 - is[i]++
 - isp[i]++
 - si llamada en zona de lb entonces
 - js[i]++
 - jsz[i][ve]++

- si (bt[i]== C) && (btotal < NUMCELL*C) entonces
 - si (llamada en zona de lb) && (aux < B) && (bt[ve] < C) entonces
 - is[i]++
 - js[i]++
 - jsz[i][ve]++
 - jjs[i][ve]++
 - si (llamada en zona de lb) && (directo) && ((aux>=B) || (bt[ve]>=C))
 - token_kill
 - si (llamada en zona de lb) && (directo=NO) && ((aux>=B) || (bt[ve]>=C))
 - si (i==0)
 - si hay canales propios en zona de lb de vebuscado entonces
 - is[i]++
 - isp[i]++
 - js[i]++
 - jsz[i][ve]++
 - saldodecac=YES
 - salgodecac2=YES
 - canalpropio=YES
 - si no
 - token_kill
 - si no
 - token_kill
 - si (llamada fuera de la zona de lb) && (directo=YES)
 - token_kill
 - si (llamada fuera de la zona de lb) && (directo=NO)
 - si hay canales propios en zona de lb de vebuscado entonces
 - is[i]++
 - isp[i]++
 - saldodecac=YES
 - salgodecac2=NO
 - si no
 - token_kill
- si no
 - token_kill

El pseudocódigo de la función *simular* se expone a continuación:

- $serv_m[i]$, $arrv_m[i]$ = medias de tasa de servicio y tasa de llamadas en célula i
 - $serv_s[i]$, $arrv_s[i]$ = desviación estándar de tasa de servicio y tasa de llamadas en célula i
 - $serv_d[i]$, $arrv_d[i]$ = tipo de distribución de tasa de servicio y tasa de llamadas en célula i
 - sim_duratn = tiempo de simulación
 - $cell_OC[i]$ = llamadas ofrecidas en célula i
 - $cell_LC[i]$ = llamadas perdidas en célula i
 - $cell_AC[i]$ = llamadas aceptadas en célula i
 - $next_call_t$ = tiempo de la siguiente llamada
 - $call_st$ = duración de la llamada
 - $call_lbt$, = duración de residencia en la zona de load balancing
 - $call_nolbt$ = duración de residencia fuera de la zona lb
 - tk_time_lf = tiempo restante de llamada
 - $PP[i]$ = probabilidad de pérdidas en la célula i
-
- inicialización de tokens: $token_init$
 - para $i \leftarrow 0$ hasta $NUMCELL$ hacer
 - generación nuevo token: $token_new$
 - planificación evento nueva llamada: $schedule(nueva\ llamada, 0)$
 - planificación evento transitorio: $schedule(evento\ transitorio, tiempo\ transitorio)$
 - mientras (tiempo de simulación > tiempo actual) hacer
 - extracción del evento de tabla de eventos: $cause(event)$
 - seleccionar event
 - caso evento nueva llamada
 - planificación evento nuevo servicio: $schedule(nuevo\ servicio, 0)$
 - generación nuevo token
 - seleccionar $arrv_d[i]$
 - caso expon
 - $next_call_t = expntl(arrv_m[i])$
 - caso erlng y caso hyper del mismo modo
 - planificación evento nueva llamada: $schedule(nueva\ llamada, next_call_t)$
 - caso evento nuevo servicio
 - $cell_OC[i]++$
 - si control de acceso entonces
 - $cell_AC[i]++$
 - seleccionar $serv_d[i]$
 - caso expon
 - $call_st = expntl(serv_m[i])$
 - $call_lbt = expntl(serv_m[i]/p)$
 - $call_nolbt = expntl(serv_m[i]/(1-p))$
 - caso erlng y caso hyper del mismo modo
 - si llamada en zona de lb: $tk_zone_lb[tk]$
 - si no MOBTY entonces

- call_lbt=call_st
 - x= call_st - call_lbt
 - si (x<=0) entonces
 - planificación evento fin llamada: schedule(fin llamada,call_st)
 - tk_time_lf=0
 - si no
 - planificación evento fin llamada: schedule(fin llamada,call_lbt)
 - tk_time_lf=x
- si no
 - si no MOBTY entonces
 - call_nolbt=call_st
 - x= call_st - call_nolbt
 - si (x<=0) entonces
 - planificación evento fin llamada: schedule(fin llamada,call_st)
 - tk_time_lf=0
 - si no
 - planificación evento fin llamada: schedule(fin llamada,call_nolbt)
- si salgodecac
 - si salgodecac2=NO entonces
 - si canalpropio
 - token_kill
 - isp[i]--
 - jjs[i][vebuscado]++
 - canal_propio [tkkill]=NO
 - si salgodecac2 entonces
 - si canalpropio
 - token_kill
 - isp[i]--
 - jjs[i][vebuscado]++
 - canal_propio [tkkill2]=NO
- si no
 - cell_LC[i]++
- caso evento fin llamada
 - seleccionar serv_d[i]
 - caso expon
 - call_lbt=expntl(serv_m[i]/p)
 - call_nolbt=expntl(serv_m[i]/(1-p))
 - caso erlng y caso hyper del mismo modo
 - si (tk_time_lf >0) entonces
 - si zona lb
 - si canalpropio=NO
 - is[i]--
 - js[i]--
 - jsz[i][ve]--

- jjs[i][ve]--
- is[ve]++
- isp[ve]++
- cell_n[tk]=ve
- zone_lb[tk]=NO
- si (i=0)
 - ve=0
- si no
 - ve=(int)(ranf()*(NUMCEL
L-1))+1;
- canalpropio=YES
- x=time_lf[tk] - call_nolbt
- if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel,
time_lf[tk])
- si no
 - time_lf[tk] = x
 - schedule(end_dwel,
call_nolbt)
- si canalpropio=YES
 - si bt[ve]<C
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - is[ve]++
 - isp[ve]++
 - cell_n[tk]=ve
 - zone_lb[tk]=NO
 - si (i=0)
 - ve=0
 - si no
 - ve=(int)(ranf()*(NU
MCELL-1))+1;
 - canalpropio=YES
 - x=time_lf[tk] - call_nolbt
 - if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel,
time_lf[tk])
 - si no
 - time_lf[tk] = x
 - schedule(end_dwel,
call_nolbt)
 - si bt[ve]>=C
 - is[i]--
 - isp[i]--
 - js[i]--

- jsz[i][ve]--
 - token_kill(tk)
 - si zona lb=NO
 - js[i]++
 - jsz[i][ve]++
 - zone_lb[tk]=YES
 - x=time_lf[tk] - call_nolbt
 - if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel, time_lf[tk])
 - si no
 - time_lf[tk] = x
 - schedule(end_dwel, call_lbt)
- si (tk_time_lf <0) entonces
 - si zona lb
 - si ((isp[i]==0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0))
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - si (jsz[i][ve]==jjs[i][ve])
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si (jsz[i][ve]>jjs[i][ve])
 - si canalpropio=YES
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - si no
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si no
 - si ((isp[i]>0) & (jsz[i][ve]==0) & (jjs[i][ve]==0))
 - is[i]--

- isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0) & (isp[i]>jsz[i][ve]))
 - is[i]--
 - isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0) & (is[i]>jsz[i][ve]))
 - is[i]--
 - isp[i]--
- caso evento transitorio
 - cell_OC[i]=0
 - cell_LC[i]=0
 - cell_AC[i]=0
- $PP[i]=(cell_LC[i])/(cell_OC[i])$

7.4. Descripción del método de simulación en un sistema multicelular compuesto por 3 anillos.

El pseudocódigo del control de acceso de canal en un sistema multicelular es el siguiente:

- ve = célula vecina elegida
 - is[i] = número de llamadas en curso en la célula i (ya sean canales propios o prestados de otras células)
 - isp[i] = número de llamadas en curso que utilizan canales propios de la célula i
 - js[i] = número de llamadas en curso en zona lb de la célula i (ya sean propios o prestados de otras células)
 - jsz[i][ve] = número de llamadas en curso en zona lb de la célula i que comparte la zona de lb de la célula ve (ya sean propios o prestados de ve)
 - jjs[i][ve] = número de canales que la célula i toma prestados de la célula ve
 - bt[i] = número de canales utilizados en la célula i ya sean canales propios o canales que está prestando a otras células
 - C = número de canales en cada célula
 - B = número de canales prestables en cada célula
 - p = probabilidad de que un nuevo usuario participe en un balance de carga
 - NUMCELLTOTAL = número de células de los 3 anillos
 - NUMCELLCLU = número de células de cada uno de los clusters que componen el sistema multicelular
 - NUMCELLCLUS = número máximo de células de cada uno de los clusters que componen el sistema multicelular
-
- NUMCELLTOTAL=37
 - NUMCELLCLUS=7
-
- Para $0 \leftarrow i$ hasta (NUMCELLTOTAL -1) hacer
 - vecino[i][0]
 - vecino[i][1]
 - vecino[i][2]
 - vecino[i][3]
 - vecino[i][4]
 - vecino[i][5]
-
- Si (NUMCELLTOTAL==37)
 - Para $0 \leftarrow i$ hasta (NUMCELLTOTAL -1) hacer
 - Para $0 \leftarrow j$ hasta (NUMCELLCLUS -1) hacer
 - si (vecino[i][j]!=-1)
 - $bt[i]=bt[i]+jjs[vecino[i][j]][i]$

- $bt[i]=bt[i]+isp[i]$
- NUMCELLCLU=1
- Para $0 \leftarrow i$ hasta (NUMCELLCLUS -1) hacer
 - si (vecino[cell][i]!=-1)
 - si $(1/arrv_m[vecino[cell][i]]!=0)$
 - NUMCELLCLU=NUMCELLCLU+1
- $ve=vecino[i][(int)(ranf()*(NUMCELLCLU-1))]$
- si $(ranf()<=(NUMCELLCLU-1)*p)$
 - zona lb = YES
- si no
 - zona lb = NO
- si $(0 < bt[i] < C) \ \&\& \ (bttotal < NUMCELLCLU*C)$ entonces
 - canalpropio=YES
 - $is[i]++$
 - $isp[i]++$
 - si llamada en zona de lb entonces
 - $js[i]++$
 - $jsz[i][ve]++$
- si $(bt[i]== C) \ \&\& \ (bttotal < NUMCELLCLU*C)$ entonces
 - si (llamada en zona de lb) $\&\& \ (aux < B) \ \&\& \ (bt[ve] < C)$ entonces
 - $is[i]++$
 - $js[i]++$
 - $jsz[i][ve]++$
 - $jjs[i][ve]++$
 - si (llamada en zona de lb) $\&\& \ (directo) \ \&\& \ ((jjs[i][ve]>=B) \ || \ (bt[ve]>=C))$
 - token_kill
 - si (llamada en zona de lb) $\&\& \ (directo=NO) \ \&\& \ ((jjs[i][ve]>=B) \ || \ (bt[ve]>=C))$
 - si hay canales propios en zona de lb de vebuscado entonces
 - $is[i]++$
 - $isp[i]++$
 - $js[i]++$
 - $jsz[i][ve]++$
 - saldodecac=YES
 - salgodecac2=YES
 - canalpropio=YES
 - si no
 - token_kill
 - si (llamada fuera de la zona de lb) $\&\& \ (directo=YES)$
 - token_kill
 - si (llamada fuera de la zona de lb) $\&\& \ (directo=NO)$
 - si hay canales propios en zona de lb de vebuscado entonces
 - $is[i]++$
 - $isp[i]++$

- saldodecac=YES
 - salgodecac2=NO
 - si no
 - token_kill
- si no
 - token_kill

El pseudocódigo de la función *simular* se expone a continuación:

- $serv_m[i]$, $arrv_m[i]$ = medias de tasa de servicio y tasa de llamadas en célula i
 - $serv_s[i]$, $arrv_s[i]$ = desviación estándar de tasa de servicio y tasa de llamadas en célula i
 - $serv_d[i]$, $arrv_d[i]$ = tipo de distribución de tasa de servicio y tasa de llamadas en célula i
 - sim_duratn = tiempo de simulación
 - $cell_OC[i]$ = llamadas ofrecidas en célula i
 - $cell_LC[i]$ = llamadas perdidas en célula i
 - $cell_AC[i]$ = llamadas aceptadas en célula i
 - $next_call_t$ = tiempo de la siguiente llamada
 - $call_st$ = duración de la llamada
 - $call_lbt$, = duración de residencia en la zona de load balancing
 - $call_nolbt$ = duración de residencia fuera de la zona lb
 - tk_time_lf = tiempo restante de llamada
 - $PP[i]$ = probabilidad de pérdidas en la célula i
-
- inicialización de tokens: $token_init$
 - para $i \leftarrow 0$ hasta $NUMCELLTOTAL-1$ hacer
 - generación nuevo token: $token_new$
 - planificación evento nueva llamada: $schedule(nueva\ llamada, 0)$
 - planificación evento transitorio: $schedule(evento\ transitorio, tiempo\ transitorio)$
 - mientras (tiempo de simulación > tiempo actual) hacer
 - extracción del evento de tabla de eventos: $cause(event)$
 - seleccionar event
 - caso evento nueva llamada
 - planificación evento nuevo servicio: $schedule(nuevo\ servicio, 0)$
 - generación nuevo token
 - seleccionar $arrv_d[i]$
 - caso expon
 - $next_call_t = expntl(arrv_m[i])$
 - caso erlng y caso hyper del mismo modo
 - planificación evento nueva llamada: $schedule(nueva\ llamada, next_call_t)$
 - caso evento nuevo servicio
 - $cell_OC[i]++$
 - si control de acceso entonces
 - $cell_AC[i]++$
 - seleccionar $serv_d[i]$
 - caso expon
 - $call_st = expntl(serv_m[i])$
 - $call_lbt = expntl(serv_m[i]/p)$
 - $call_nolbt = expntl(serv_m[i]/(1-p))$
 - caso erlng y caso hyper del mismo modo
 - si llamada en zona de lb: $tk_zone_lb[tk]$
 - si no MOBTY entonces

- call_lbt=call_st
 - x= call_st - call_lbt
 - si (x<=0) entonces
 - planificación evento fin llamada: schedule(fin llamada,call_st)
 - tk_time_lf=0
 - si no
 - planificación evento fin llamada: schedule(fin llamada,call_lbt)
 - tk_time_lf=x
- si no
 - si no MOBTY entonces
 - call_nolbt=call_st
 - x= call_st - call_nolbt
 - si (x<=0) entonces
 - planificación evento fin llamada: schedule(fin llamada,call_st)
 - tk_time_lf=0
 - si no
 - planificación evento fin llamada: schedule(fin llamada,call_nolbt)
- si salgodecac
 - si salgodecac2=NO entonces
 - si canalpropio
 - token_kill
 - isp[i]--
 - jjs[i][vebuscado]++
 - canal_propio [tkkill]=NO
 - si salgodecac2 entonces
 - si canalpropio
 - token_kill
 - isp[i]--
 - jjs[i][vebuscado]++
 - canal_propio [tkkill2]=NO
- si no
 - cell_LC[i]++
- caso evento fin llamada
 - seleccionar serv_d[i]
 - caso expon
 - call_lbt=expntl(serv_m[i]/p)
 - call_nolbt=expntl(serv_m[i]/(1-p))
 - caso erlng y caso hyper del mismo modo
 - si (tk_time_lf >0) entonces
 - si zona lb
 - si canalpropio=NO
 - is[i]--
 - js[i]--
 - jsz[i][ve]--

- jjs[i][ve]--
- is[ve]++
- isp[ve]++
- cell_n[tk]=ve
- zone_lb[tk]=NO
- invecinoeleg=(int)(ranf()*(NUMCELL CLU-1))
- ve=vecino[cell][invecinoeleg]
- canalpropio=YES
- x=time_lf[tk] - call_nolbt
- if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel, time_lf[tk])
- si no
 - time_lf[tk] = x
 - schedule(end_dwel, call_nolbt)
- si canalpropio=YES
 - si bt[ve]<C
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - is[ve]++
 - isp[ve]++
 - cell_n[tk]=ve
 - zone_lb[tk]=NO
 - invecinoeleg=(int)(ranf()*(NUMCELLCLU-1))
 - ve=vecino[cell][invecinoeleg]
 - canalpropio=YES
 - x=time_lf[tk] - call_nolbt
 - if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel, time_lf[tk])
 - si no
 - time_lf[tk] = x
 - schedule(end_dwel, call_nolbt)
 - si bt[ve]>=C
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - token_kill(tk)
- si zona lb=NO

- js[i]++
- jsz[i][ve]++
- zone_lb[tk]=YES
- x=time_lf[tk] - call_nolbt
- if (x<=0)
 - time_lf[tk] = 0.0
 - schedule(end_dwel, time_lf[tk])
- si no
 - time_lf[tk] = x
 - schedule(end_dwel, call_lbt)
- si (tk_time_lf <0) entonces
 - si zona lb
 - si ((isp[i]==0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0))
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0))
 - si (jsz[i][ve]==jjs[i][ve])
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si (jsz[i][ve]>jjs[i][ve])
 - si canalpropio=YES
 - is[i]--
 - isp[i]--
 - js[i]--
 - jsz[i][ve]--
 - si no
 - is[i]--
 - jjs[i][ve]--
 - js[i]--
 - jsz[i][ve]--
 - si no
 - si ((isp[i]>0) & (jsz[i][ve]==0) & (jjs[i][ve]==0))
 - is[i]--
 - isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]==0) & (isp[i]>jsz[i][ve]))

- is[i]--
 - isp[i]--
 - si ((isp[i]>0) & (jsz[i][ve]>0) & (jjs[i][ve]>0) & (is[i]>jsz[i][ve]))
 - is[i]--
 - isp[i]--
 - caso evento transitorio
 - cell_OC[i]=0
 - cell_LC[i]=0
 - cell_AC[i]=0
- $PP[i]=(cell_LC[i])/(cell_OC[i])$