



PARALELIZACIÓN DE UN ALGORITMO DE CÁLCULO DE LA RESPUESTA IMPULSIONAL DE UN CANAL IR DIFUSO EN INTERIORES

Elsa Macías¹, Alvaro Suárez¹, Rafael Pérez², F.J. López Hernández³

¹GAC (Grupo de Arquitectura de Computadores), Dpto. de Electrónica, Telemática y Automática

²GPDS (Grupo de Procesado Digital de la Señal), Dpto. de Señales y Comunicaciones
Universidad de Las Palmas de G.C., e-mail: {alvaro, rperez}@cic.teleco.ulpgc.es

³Dpto. de Tecnología Fotónica, Universidad Politécnica de Madrid, email: dxin@tfo.upm.es

En los últimos años ha habido una explosión de la utilización de los sistemas de comunicación no cableados. En este tipo de sistemas, los dispositivos de comunicación pueden ser desde simples emisores y receptores hasta sistemas complejos basados en computadores. Las redes de área local basadas en computadores personales portátiles son un hecho tangible. Una forma de conectar estos sistemas es utilizar comunicación infrarroja. La simulación de la respuesta impulsional de un canal de infrarrojos en un laboratorio es costosa, por ello se recurre al uso de computadores. Sin embargo, debido a la gran cantidad de tiempo necesario para obtener resultados, en algunos casos el empleo de los computadores se puede hacer inviable. Una forma de resolver este problema es ejecutar el programa de simulación en varios procesadores, en paralelo. En este artículo presentamos la simulación en un sistema multicomputador para poder obtener respuestas en un tiempo razonable.

1. INTRODUCCIÓN

En la actualidad existen sistemas que integran equipos terminales conectados entre ellos mediante un sistema de comunicación. En algunos casos estos equipos terminales pueden ser simples equipos emisores o receptores de señal y en otros casos pueden estar empujados en computadores. En cualquier caso, un elemento importante es el tipo de interconexión entre equipos a usar. El método tradicional de interconexión es el cableado. Sin embargo, el uso masivo de computadores portátiles y la comunicación de grandes cantidades de datos, han producido la aparición de las redes de área local no cableadas (denominadas **WLAN**, de las siglas en inglés Wireless Local Area Network). La portadora empleada en el diseño de estas redes puede ser de radiofrecuencia (**RF-WLAN**, del inglés Radio Frequency-Wireless Local Area Network, o redes de área local no guiadas por radiofrecuencia), o por infrarrojo (denominadas **IR-WLAN**, de las siglas en inglés Infrared-Wireless Local Area Network, o redes de área local no guiadas por infrarrojo).

Los sistemas de comunicación no cableados presentan varias ventajas sobre los interconectados por cable. Entre éstas están: la flexibilidad para recolocar equipos terminales y su facilidad de movilidad dentro del entorno en el que se puede emitir y recibir señales [1].

Los *enlaces por infrarrojo* proporcionan comunicaciones no cableadas de alta velocidad sobre distancias relativamente cortas. En la actualidad estas comunicaciones están recibiendo un interés creciente debido a que presentan algunas ventajas sobre las *comunicaciones vía radio*. Ejemplos de estas ventajas son: no necesitan licencia gubernativa, son inmunes a las interferencias producidas por el ruido electromagnético y los componentes empleados (emisores y receptores de infrarrojos que operan a alta velocidad), son baratos y se pueden implantar en hardware de forma sencilla. Por otro lado, las comunicaciones vía infrarrojo están confinadas dentro del entorno cerrado en el que se originan, lo que asegura la privacidad de las comunicaciones y evita interferencias con enlaces que operan en otras habitaciones.

Las comunicaciones mediante canales infrarrojos también presentan algunas desventajas. Entre ellas están: distancia de utilización limitada y no son inmunes a la luz ambiente ni al alumbrado fluorescente e incandescente que inducen ruido sobre el receptor de luz infrarroja. Estas fuentes de ruido están presentes en muchos entornos cerrados como oficinas, empresas, hogares, etc. Además, la presencia de obstáculos en la línea de emisión impide la comunicación de enlaces punto a punto y, por otra parte, la comunicación desde una habitación a otra necesita puntos de acceso de infrarrojos.

Los enlaces infrarrojos se clasifican atendiendo al grado de direccionalidad del transmisor y del receptor y, a la existencia o ausencia de visión directa (**LOS**, del inglés Line Of Sight o **DBC**, de las siglas Directive Beam Configuration) entre el emisor y el receptor [2]. En el primer caso se trata de enlaces directos o no directos según se empleen transmisores y receptores direccionales o de ángulo ancho. Los enlaces no directos se emplean en la comunicación de terminales móviles. En los enlaces directos, la potencia recibida es máxima debido a que se reduce el camino de atenuación de la señal y la recepción de ruido generado por la luz ambiente. También es posible establecer enlaces híbridos que emplean transmisores y receptores con diferentes grados de direccionalidad. El diseño de un enlace LOS maximiza la potencia recibida y disminuye la dispersión por multitrayectoria (multipath dispersion). La dispersión por multitrayectoria se produce debido a las múltiples reflexiones que sufre la señal emitida hasta alcanzar al receptor.



Un canal de propagación infrarrojo en interiores es, esencialmente, un canal con propagación multitrayectoria. Esto significa que si se emite un pulso de potencia óptica de duración muy pequeña, como una función delta de Dirac, la señal recibida en el receptor estaría constituida por un tren de impulsos con diferentes retardos y amplitudes. Este resultado se debe a las múltiples reflexiones que sufre el pulso emitido en las paredes, muebles, personas y cualquier otro objeto situado en el lugar en el que están los emisores y receptores.

Para simular por computador la respuesta impulsional de un canal IR difuso, se parte de un modelo para los emisores, receptores y reflectores que se caracteriza mediante magnitudes escalares y vectoriales. En [3] se presenta un modelo y un algoritmo de simulación en el que se tienen en cuenta el número de reflexiones sobre las paredes (rebotes). En [10] se presenta una mejora del algoritmo anterior introduciendo una matriz en la que se almacenan las interacciones entre las diferentes superficies reflectoras, haciendo una simulación por tiempo. Además se mejora el tiempo de ejecución aprovechando propiedades de simetrías entre las superficies reflectoras para ahorrar cálculos. En [9] se presenta un estudio de los modelos deterministas anteriores y se implanta un modelo estadístico. Con este modelo estadístico se obtiene mayor flexibilidad y tiempos de ejecución muy aceptables. En los algoritmos deterministas comentados existe un problema importante: el número de cálculos que se deben realizar para simular situaciones realistas es extremadamente elevado. Esto significa que el tiempo de ejecución podría llegar a ser de hasta años. Por tanto, estos algoritmos no son viables debido a que para el momento en el que se obtuviese la respuesta de la simulación, probablemente los dispositivos simulados ya estarían obsoletos.

Por tanto es importante reducir este tiempo de ejecución tan elevado. En este artículo se presenta un algoritmo de simulación secuencial. Este algoritmo está inspirado en el algoritmo de [10] si bien se ha programado partiendo de cero, sin aprovechar las propiedades de simetría y sin introducir la matriz que almacena las interacciones entre las superficies reflectoras. Posteriormente se paraleliza aplicando transformaciones sobre el código secuencial para obtener tiempos de ejecución relativamente razonables.

La estructura del artículo es la siguiente: En la sección siguiente presentamos el modelo del sistema, en la sección 3 presentamos los programas de simulación por computador, en la sección 4 se presentan los resultados de la simulación por computador expresados en tiempos de ejecución y eficiencia de uso de los procesadores. Por último, presentamos algunas conclusiones y líneas de trabajo futuras.

2. EL MODELO DE SIMULACIÓN

Dado un sistema que consiste en una habitación, en cuyo interior se encuentran uno o varios emisores de

luz (LEDs) y uno o varios receptores, se trata de hallar la distribución temporal de potencia en cada uno de los receptores situados en el interior de la habitación debido a la emisión de señales procedentes de los emisores situados en el interior de dicha habitación.

El emisor transmite potencia óptica en el instante de tiempo t_0 con un cierto diagrama de radiación, esto es, la potencia no se transmite por igual en todas las direcciones. Parte de esa potencia llega al receptor y otra parte alcanza a las superficies reflectoras de la habitación. La potencia que llega a las superficies reflectoras rebota hacia otros lugares de la habitación, incluso hacia el receptor. Por tanto, al receptor le llega potencia por dos caminos diferentes:

- A través de una línea recta directa hipotética trazada entre emisor y receptor, y
- A través de un camino hipotético (“emisor pared_1 >> pared_2 >> pared_3 >>...>> receptor”), generado por rebotes en las superficies reflectoras (paredes de la habitación).

La distribución temporal de la potencia que llega al receptor está formada por un pico de potencia correspondiente al enlace directo entre emisor y receptor, seguido de una cola de potencia debida a los rebotes en las paredes de la habitación. La potencia en el receptor disminuye a medida que avanza el tiempo debido a los múltiples rebotes que van atenuando a la señal.

Para considerar las reflexiones en las paredes de la habitación éstas se dividen en elementos discretos a los que denominamos celdas. En la figura 1 se muestran las paredes de una habitación dividida en celdas mediante el trazado de líneas rectas perpendiculares y paralelas a alguna de las aristas que las conforman.

Una celda, con potencia distinta de cero ilumina al resto de celdas que están en paredes diferentes a la suya. El tiempo que tarda en iluminar una celda emisora a una celda receptora, se calcula como d/c donde: c es la velocidad de la luz y d es la distancia que existe entre ambas celdas.

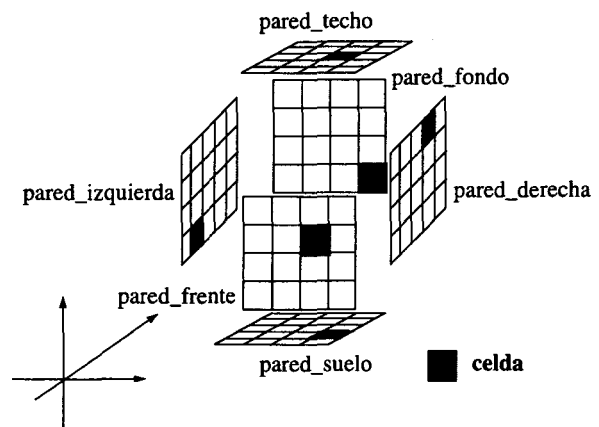


Figura 1. División en celdas de las paredes de una habitación.

La potencia existente en una celda en el instante de tiempo t_i , depende de las potencias emitidas por otras celdas y emisores en instantes de tiempo t_j anteriores ($j < i$). De igual manera, la potencia que llega al receptor en t_k depende de la potencia emitida por las celdas y emisores en tiempos t_l inferiores a t_k ($l < k$). Por tanto, debido a la dependencia temporal que existe en el cálculo de la potencia que llega a una celda o a un receptor, se emplea una simulación dirigida por tiempo.

En la figura 2 se muestra una habitación con un emisor y un receptor. Se dibuja la línea de visión directa entre los dos elementos. También se dibujan los rebotes producidos en varias celdas. Nótese que para cada instante de simulación se debe averiguar qué celdas están iluminadas en ese instante; sin embargo este cálculo es fuertemente dependiente de las celdas que estaban iluminadas en los instantes de tiempo anteriores.

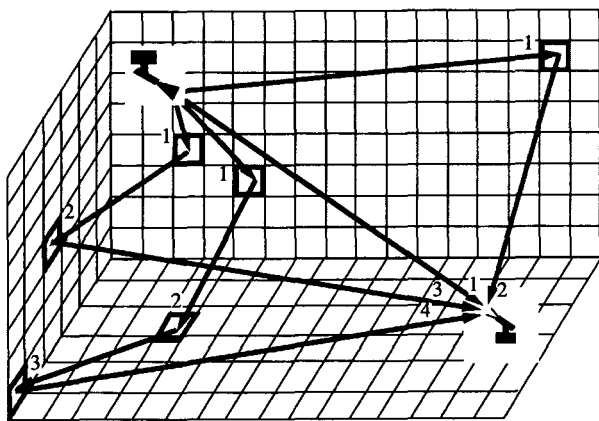


Figura 2. Visión directa y hasta tres rebotes.

Los emisores y receptores se representan mediante las siguientes magnitudes vectoriales y escalares [3]: vector de posición, vectores unitarios que determinan sus orientaciones (perpendiculares a la superficie de cada elemento), potencia óptica de salida (emisor), modo de emisión (emisor), diagrama de emisión lambertiano (emisor), área activa (receptor) y campo de visión (receptor). Las características reflectoras de cualquier superficie dependen del material y de la textura de la superficie. El modelo del reflector correcto, para la mayoría de las superficies reflectoras, debe considerar dos componentes [4]: una no direccional o difusa según un perfil de radiación determinado, y otra direccional o especular. Sin embargo, una buena aproximación para el modelo del reflector consiste en considerar solamente las componentes difusas, con un diagrama de emisión lambertiano ideal, esto es, el modo de emisión de la fuente es igual a 1 [3]. Este modelo simplificado se conoce con el nombre de Modelo de reflexión lambertiano.

La potencia recibida y el tiempo en el que ésta llega al receptor, dependen de la distancia d entre el emisor y el receptor (visión directa) o entre una celda y el receptor. La distancia d se calcula considerando la posición del emisor y el receptor (o la celda y el receptor).

3. SIMULACIÓN POR COMPUTADOR

Para simular la respuesta al impulso de luz o señal generada por un emisor de luz en una habitación vacía y siguiendo el modelo anterior, se utilizan estructuras de datos para representar los elementos que constituyen el modelo. Además se utiliza una matriz bidimensional, cuyo número de filas está dado por la siguiente expresión: Donde t_{\max} y t_{\min} son el tiempo máximo y mínimo que transcurre para que una celda emisora ilumine a una celda receptora respectivamente y, d_{\max} y d_{\min} son las distancias máxima y mínima respectivamente entre cualesquiera dos celdas de la habitación. El número de columnas de la matriz de potencias es igual al número de celdas total de la habitación. Esta matriz la denominamos **m_pot** y las filas representan instantes discretos de simulación, mientras que las columnas representan a cada una de las celdas. Cada elemento de la matriz almacena un valor de potencia producido por la iluminación debida a las celdas o a los emisores.

En el apartado siguiente se describen las acciones secuenciales a realizar en el programa que se diseñó.

3.1. Simulación secuencial

En el programa secuencial básicamente se especifican un conjunto de bucles anidados. Se recorren todos los instantes discretos de simulación y para cada instante se recorren todas las paredes y sus celdas. Para cada celda se comprueba si está iluminada (valor de potencia distinto de cero). En caso afirmativo, se recorren todas las celdas pertenecientes al resto de las paredes (sin considerar la pared a la que pertenece la celda que se está analizando: $\text{pared}_d \neq \text{pared}_o$), calculando el efecto y el retardo que existe en la propagación de la señal procedente de la celda iluminada. Para ello hay que tener en cuenta las posiciones de las celdas dentro de la habitación y el diagrama de radiación lambertiano de la celda iluminada. A continuación se actualiza la matriz de potencias (**actualizar m_pot**). La actualización de esta matriz consiste en acceder a una fila superior al instante de tiempo discreto de simulación que se está considerando (d), y acceder a la columna marcada por la celda a la cual se quiere comprobar si le llega potencia. En ese elemento se almacena el valor recién calculado. El acceso a las filas de **m_pot** se hace en módulo n (número de filas de dicha matriz).

Por cada celda iluminada también se comprueba si existe algún receptor iluminado por esa celda. En caso afirmativo, se actualiza la potencia que llega al receptor debido a la iluminación de dicha celda.

El proceso anterior se repite para todas las celdas de la habitación. Acabado este proceso, entonces se pone a cero todos los elementos de la fila d de **m_pot**. Esto es necesario porque en la siguiente iteración de d podría ser posible que se utilizaran algunos de los elementos de esa fila para acumular el valor de la potencia en las celdas.

Por tanto si no se pone previamente a cero entonces se podrían calcular valores incorrectos de la potencia.

A continuación, si se han recorrido todas las filas de m_{pot} , se actualiza la potencia que envía cada uno de los emisores de señal a las celdas en las que se han dividido las paredes de la habitación.

Cuando finaliza esta actualización, se procede con el siguiente valor del tiempo de simulación discreto.

```

for d (1, fin_tiempo_simulación, 1)
  for pared_o (1, max_num_paredes, 1)
    for celda_o (1, fin_pared_o, 1)
      if celda_o iluminada then
        for pared_d (1, max_num_paredes (pared_d<>pared_o), 1)
          for celda_d (1, fin_pared_d, 1)
            Calcular efecto y retardo entre celda origen y destino
            Calcular potencia y actualizar m_pot
          end_for
        end_for
      for receptor (0, max_num_receptores-1, 1)
        if receptor iluminado por celda_o then
          Calcular potencia entre celda origen y receptor
        end_if
      end_for
    end_if
  end_for
end_for
Actualizar a cero la fila leída de la matriz m_pot
if recorrida m_pot then
  for emisor(0, max_num_emisores-1, 1)
    if emisor señal then
      Repeat para cada fila de m_pot
        for pared (1, max_num_paredes, 1)
          for celda (1, fin_pared, 1)
            if celda iluminada por emisor then
              Calcular potencia emisor-celda.
            end_if
          end_for
        end_for
      end_repeat
    end_if
  end_for
end_if
end_for

```

Figura 3. Algoritmo secuencial.

En la figura 3 se muestra un pseudocódigo. Se puede comprobar que el código básicamente está constituido por un conjunto de bucles anidados. En estos bucles se actualizan variables vectoriales cuya estructura es relativamente sencilla.

El número de operaciones que se realizan en este algoritmo es muy elevado. Por cada celda iluminada se ha de recorrer el resto de las celdas de la habitación. Y esto se repite para cada instante de simulación. Además, para cada operación se tienen que realizar cálculos costosos en tiempo de ejecución (cálculos de ángulos, distancias, proyecciones, etc.) con números almacenados utilizando la representación en punto flotante (lo que significa que las operaciones en el procesador aún serán más lentas que si se utilizan números en punto fijo). Por

tanto estamos frente a un problema realmente complejo, en el que a medida que el número de celdas aumenta estaríamos frente a posibles tiempos de ejecución de incluso años [3].

Una forma de reducir este elevado coste computacional es repartir el número de cálculos entre diferentes procesadores. Si los procesadores pudieran trabajar al mismo tiempo entonces reduciríamos el tiempo de ejecución considerablemente a medida que empleamos más procesadores.

3.2. Paralelización del programa secuencial

El objetivo principal de la paralelización de códigos secuenciales consiste en obtener una nueva ejecución de los cálculos secuenciales de tal forma que algunos de estos se puedan realizar de forma paralela.

Para obtener un programa paralelo, paralelizando un programa secuencial, es necesario particionar y distribuir el código secuencial y los datos entre los procesadores y memorias de la máquina paralela [6]. En los últimos años se han construido potentes máquinas que disponen de más de un procesador (arquitecturas paralelas) y su complejidad se ha incrementado enormemente. La ejecución eficiente de los programas en estas máquinas requiere que se tomen en cuenta las características particulares de las mismas. En la actualidad no existe una herramienta automática que permita realizar el particionado de código y datos entre procesadores. En ausencia de estas herramientas de particionado y distribución, el código y los datos deben ser particionados por el programador especializado.

Los bucles anidados son el núcleo de aplicaciones científicas y de ingeniería que transforman estructuras constituidas por un elevado número de datos. En estos bucles se efectúan una gran cantidad de cálculos y accesos a memoria concentrándose en este tipo de código un gran porcentaje del tiempo de ejecución de estos programas. Por tanto, es en este tipo de códigos en los que se centra el programador para paralelizar los códigos secuenciales transformando el código secuencial para aprovechar la posible ejecución paralela de los cálculos a realizar, a estas transformaciones se les suele denominar reestructuración de código [5].

En el código secuencial de la figura 3 existen varios bucles en los que se efectúan la mayor parte de los cálculos. Para paralelizar este código es preciso centrar nuestra atención en estos bucles. Para paralelizar el programa mediante reestructuración de código, teniendo en cuenta los bucles, hemos de realizar las siguientes fases:

1. *Distribución de datos.* Dado que los bucles que recorren las celdas son los que influyen definitivamente en el tiempo de ejecución, entonces nos centramos en cómo distribuir las celdas entre los procesadores. El objetivo es conseguir que los procesadores realicen los

cálculos del efecto y retardo entre conjuntos de celdas disjuntos. Esto conlleva el que deben trabajar con columnas diferentes de la matriz m_{pot} .

2. *Distribución de cálculos.* Para conseguir el objetivo del paso anterior se debe modificar el código. Esto es, debemos obtener el código que ejecutarían los procesadores. La forma más sencilla de hacer esto es reestructurar el código secuencial haciendo que los procesadores ejecuten el mismo código, pero parametrizándolo mediante variables que identifiquen las operaciones que ejecuta cada procesador.

Para llevar a cabo este paso se deben aplicar transformaciones de código. Una técnica de paralelización de códigos secuenciales es la denominada *Block Scheduling Equilibrado* [6]. Si cada procesador dispone de una variable local denominada my_id que le identifica unívocamente, entonces con dos vectores denominados $imin$ e $imax$ se pueden obtener los límites inferiores ($imin[my_id]$) y superiores ($imax[my_id]$) de los bucles tal que el procesador my_id trabaje con un conjunto de celdas disjunto al del resto de los procesadores.

3. *Paso de mensajes.* Con los pasos anteriores, cada procesador necesita datos que no conoce puesto que han sido calculados por los otros procesadores. Concretamente, en nuestro caso, es necesario que todos los procesadores comuniquen la parte de la matriz de potencia que actualizan para que el resto de los procesadores pueden utilizar estos datos para calcular la potencia que envía su conjunto de celdas hacia el resto de celdas y receptores.

En la figura 4 se muestra el pseudocódigo del programa paralelo resultante. Se puede apreciar que un procesador, después de comprobar todas las celdas que le han sido asignadas dentro de la habitación, realiza una comunicación colectiva en la que se radía la información anteriormente mencionada.

```

for d (1, fin_tiempo_simulación, 1)
  for pared_o (1, max_num_paredes, 1)
    for celda_o (imin[pared_o], imax[pared_o], 1)
      ...
      Actualizar a cero la fila leída de la matriz m_pot
      Comunicar la siguiente fila de la matriz m_pot
      if recorrida m_pot then
        ...
        for pared (1, max_num_paredes, 1)
          for celda (imin[pared_o], imax[pared_o], 1)
            if celda iluminada por emisor then
              Calcular potencia emisor-celda.
            end_if
          end_for
        end_for
      end_if
    end_for
  end_for
end_for

```

Figura 4. Algoritmo Paralelo.

En el código resultante los procesadores efectúan una gran cantidad de cálculo y una vez acabados esos cálculos se efectúa la comunicación. Por cada iteración

del bucle d existe la necesidad de efectuar comunicaciones. Además, si el número de celdas es elevado, los mensajes a comunicar contienen una gran cantidad de datos.

4. RESULTADOS DE LA SIMULACIÓN POR COMPUTADOR

En este apartado presentamos algunos resultados de la medida de los tiempos de ejecución y de la eficiencia considerando una habitación de las siguientes características: a) dimensiones 5X5X3 metros cúbicos, b) posición del emisor en (2.5, 2.5, 3), el receptor en (4.5, 1, 0), y c) tiempo de simulación de 60 ns, utilizando diferentes tamaños de celdas.

Estos algoritmos han sido implantados en el multicomputador SP-2 del CESCA (Centro de Supercomputación de Cataluña). En el multicomputador SP-2, los procesadores no comparten memoria física (memoria distribuida), y se comunican usando paso de mensajes a través de un dispositivo de comunicación de alta velocidad, el High Performance Switch (HPS). El HPS es una red de interconexión omega [7], con conmutadores de 8x8 que utiliza encaminamiento buffered-wormhole mediante conmutación de paquetes [7]. Para especificar el paso de mensajes en el programa paralelo se ha utilizado la biblioteca MPI (Message Passing Interface) [8]. La comunicación básicamente se especifica utilizando las llamadas a **MPI_Bcast** y **MPI_Allreduce** pertenecientes a la biblioteca MPI.

En la figura 5 se muestra el tiempo de ejecución del programa secuencial y del paralelo, expresado en minutos. El programa paralelo se ha ejecutado en seis procesadores.

La reducción de tiempo de ejecución conseguida por el programa paralelo es considerable. Por ejemplo, para un tamaño de celda igual a 10 centímetros se consigue una reducción aproximadamente de un sexto del tiempo de ejecución del programa secuencial. Además se puede comprobar fácilmente que el tiempo de ejecución disminuye a medida que se aumenta el tamaño de las celdas. Por lo tanto, podemos asegurar que la paralelización del programa secuencial es efectiva.

Cuando se genera un programa paralelo es deseable que los procesadores se utilicen durante todo el tiempo de ejecución del programa paralelo. Para obtener información sobre la utilización de los procesadores se utiliza una métrica denominada eficiencia [7]. La eficiencia se calcula como:

$$Eficiencia = \frac{T_{seq}}{n \times T_{par}}$$

donde: T_{seq} es el tiempo de ejecución del programa secuencial, T_{par} es el tiempo de ejecución del programa paralelo y n es el número de procesadores. La eficiencia es una figura de mérito cuyos valores extremos son:

$$0 \leq Eficiencia \leq 1$$



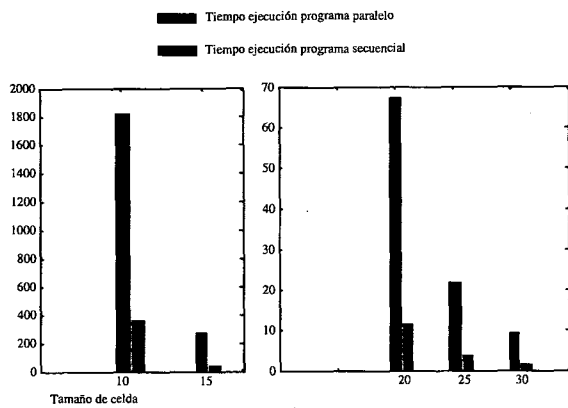


Figura 5. Tiempos de ejecución del programa secuencial y del paralelo.

En la figura 6 se muestra la eficiencia del programa paralelo para diferentes tamaños de celdas. La eficiencia tiene un valor elevado lo cual significa que los procesadores han sido utilizados durante todo el tiempo de ejecución del programa paralelo. Esto se debe a que las comunicaciones se han minimizado, haciendo que los procesadores realicen una cantidad de cálculos elevada. En las arquitecturas como el SP-2 esta técnica da buenos resultados de eficiencia.

5. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO

En las **WLAN** la portadora empleada puede ser de radiofrecuencia (**RF-WLAN**), o de infrarrojo (**IR-WLAN**). Los *enlaces por infrarrojo* proporcionan comunicaciones no cableadas de alta velocidad en distancias relativamente cortas, siendo muy apropiados para aplicaciones de interiores (habitaciones). Para simular, por computador, la respuesta de un canal de infrarrojos en interiores se parte de un modelo para los componentes emisores y receptores así como para el medio físico (canal). La simulación por computador tiene la ventaja de ser menos costosa y más flexible que la simulación real en un laboratorio. Sin embargo, el tiempo de ejecución para la simulación realista por computador puede ser muy elevado.

En este artículo hemos presentado un algoritmo secuencial que tiene un tiempo de ejecución muy elevado (de hasta 30 horas). La paralelización de este programa en un multicomputador conduce a una reducción muy considerable del tiempo de ejecución (hasta 6 horas, para el caso anterior). Se han probado varias configuraciones de emisores y receptores comprobándose la correctitud de la simulación y obteniendo una utilización eficiente de la arquitectura del multicomputador utilizado.

El trabajo presentado en este artículo, se podría ampliar considerando modelos matemáticos más complejos para los emisores. Por otro lado, se podrían considerar habitaciones en L.

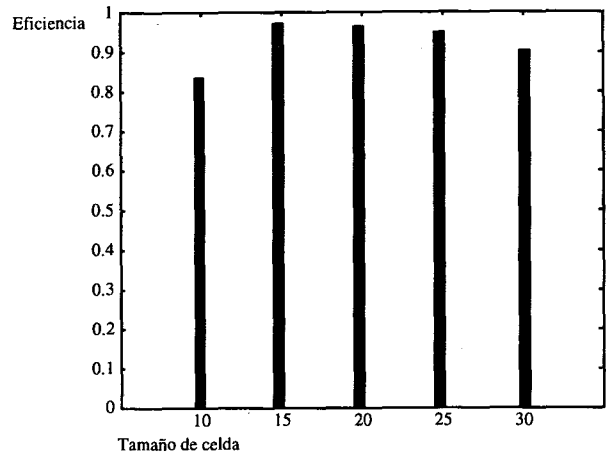


Figura 6. Eficiencia del programa paralelo.

AGRADECIMIENTOS

Agradecemos al personal del *CESCA (Centro de Supercomputación de Cataluña)* el soporte prestado en la utilización del multicomputador SP-2: sin sus máquinas ni su personal este trabajo no hubiera sido posible.

REFERENCIAS

- [1] FRITZ R. GFELLER, URS BAPST, Wireless In-House Data Communication via Diffuse Infrared Radiation, Proceedings of the IEEE, vol. 67, no. 11, pag. 1474, Noviembre 1979.
- [2] JOSEPH M. KAHN, JOHN R. BARRY, Wireless Infrared Communications, Proceedings of the IEEE, VOL. 85, NO. 2, Febrero 1997.
- [3] JOHN R. BARRY, JOSEPH M. KAHN, WILLIAN J. KRAUSE, EDWARD A. LEE, DAVID G. MESSERSCHMITT, Simulation of Multipath Impulse Response for Indoor Wireless Optical Channels, IEEE Journal on Selected areas in communications, Vol. 11, no. 3, Abril 1993.
- [4] CIPRIANO R.A.T. LOMBA, RUI T. VALADAS, A.M. DE OLIVEIRA DUARTE, Update of Propagation Losses and Impulse Response of the Indoor Optical Channel, IEEE P802.11, Septiembre 1993.
- [5] WOLFE J. M., Optimizing Compilers for Supercomputers, Pitman Publishing, London, 1989.
- [6] STEVEN BRAWER, Introduction to Parallel Programming, Academic Press, Inc., ISBN 0-12-128470-0, 1989.
- [7] HWANG KAI, Advanced Computer Architecture. Parallelism, Scalability, Programability, McGraw-Hill, ISBN-0-07-031622-8, 1993.
- [8] PETER S. PACHECO, Parallel Programming with MPI, Morgan Kaufmann Publishers, Inc., 1997.
- [9] JOSE BERGES COLMENAR, Modelo estadístico para la simulación de redes de datos por enlace óptico no guiado, Proyecto Fin de Carrera, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Las Palmas de Gran Canaria, Noviembre 1996.
- [10] F. J. LOPEZ HERNÁNDEZ, *Dustin: A new algorithm for the calculation of the impulse response in indoor IR*