$$AnchuradelMetacamino = s = l^{*}k$$

Ecuación 4-8 Anchura del metacamino

Cuando ambos Supernodos *Origen* y *Destino* están en forma canónica menor, el metacamino por ellos formados está en forma canónica menor, es decir, está compuesto por el único camino de distancia mínima definido por el encaminamiento estático y, por lo tanto, s=1.

Definimos **longitud del metacamino** como el valor medio de las longitudes de los caminos multipaso individuales que lo componen.

$$Long(P^{*}) = (1/s)\sum_{\forall s} long(MSP_s)$$

Ecuación 4-9 Longitud de un metacamino

Definimos **latencia del metacamino** como la inversa de la suma de las inversas de las latencias de cada uno de los caminos multipaso que componen el metacamino. Estas latencias son las sufridas por los mensajes que viajan por los caminos multipaso. Estas latencias inversas son, de hecho anchos de banda, y su suma es el ancho de banda del metacamino. En esta definición usamos el mismo concepto físico de adición de resistencias eléctricas en paralelo en un circuito básico que se ha usado en el capítulo 3 para definir el canal equivalente.

$$\text{Latencia}(P^{*}) = (\sum_{\forall s} Latencia(MSPs)^{-1})^{-1}$$

Ecuación 4-10 Latencia de un metacamino

La **latencia canónica** del metacamino P* es el tiempo que el mensaje de un determinado tamaño utiliza en viajar a lo largo del metacamino canónico, es decir, el tiempo de transmisión del camino mínimo.

Definimos **ancho de banda del metacamino** como la suma de los anchos de banda de los caminos multipaso que lo forman. Es el inverso de la latencia del metacamino.

$$\text{AnchodeBanda}(P^{*}) = \text{Latencia}(P^{*})^{-1} = (\sum_{\forall s} AnchodeBanda(MSPs)$$

Ecuación 4-11 Ancho de banda de un metacamino

El **ancho de banda canónico** se define como el inverso de la latencia canónica del metacamino. Esta magnitud mide el máximo número de mensajes por unidad de tiempo que el metacamino puede aceptar.

Hasta aquí el conjunto de definiciones formales que nos permiten construir conjuntos de caminos alternativos entre cada canal de la aplicación. Vamos ahora a explicar cómo se usan los metacaminos para una cierta aplicación dada, con un cierto número de canales por los que se envían mensajes. El procedimiento es asignar un metacamino P^* a cada uno de los canales de la aplicación mediante la asignación de un Supernodo Origen al nodo origen del canal y un Supernodo Destino al nodo destino del canal. Esta asignación la gestiona de forma transparente y dinámica el encaminador DRB. El Supernodo Origen entonces se comporta como un **área de dispersión de mensajes** desde el nodo origen. El Supernodo Destino funciona como un **área de recolección de mensajes** hacia el nodo destino. El metacamino es la zona por donde viajan los mensajes entre cada una de las áreas. Entonces, para cada mensaje que la tarea origen envía, se selecciona un camino multipaso MSP (*Origen*, $N_i^{SOrigen}$, N_j^{SDest}, *Destino*) perteneciente al metacamino P* y el mensaje se envía a través de él. Esta es la tarea del encaminador DRB.

Bajo este esquema, la comunicación entre fuente y destino puede verse como si se estuviera usando un ancho metacamino multicarril entre un Supernodo fuente y un Supernodo destino de mayor ancho de anda potencial que el camino original. Este camino multicarril puede entenderse como una "autopista" y las áreas de dispersión y recolección de mensajes pueden ser vistas como las áreas de entrada y salida de la autopista, respectivamente. Cada mensaje recorre su viaje a través de un camino multipaso. Cada uno de los pasos individuales, se recorre usando encaminamiento estático mínimo. Cuando se llega al primer destino intermedio, se continúa hacia el segundo, y desde el segundo hasta el destino final. Este es el encaminamiento DRB.

En una misma aplicación, varios Metacaminos pueden solaparse y usar encaminadores en común. No hay ninguna restricción en este sentido. Asimismo, varios caminos multipaso de un mismo metacamino pueden tener partes solapadas y compartir algunos de los enlaces que utilizan, pero como los enlaces no son utilizados de manera simultánea en el tiempo, se consigue un incremento del ancho de banda efectivo para ese metacamino.

Hasta ahora hemos visto como crear caminos alternativos entre fuente y destino para cada canal. Esta técnica es una metodología sistemática para construir Supernodos a partir de la topología y el ancho de anda y alargamiento del camino deseados. Hemos visto que un Supernodo no es un conjunto de nodos arbitrario o desestructurado. DRB

establece métodos uniformes cuyos objetivos son construir Supernodos sistemáticamente para cada topología y para calcular los parámetros de los metacaminos. Esta es una información estática existente en los encaminadores DRB.

Ahora debemos responder a la pregunta ¿Cómo se selecciona un determinado Supernodo? La respuesta la da la segunda parte de DRB. Esta segunda parte actúa en tiempo de ejecución, es decir, cuando se están usando los metacaminos y los caminos multipaso. Esta parte, llevada acabo por el encaminamiento bajo DRB, se encarga de decidir qué tamaño deben tener los metacaminos de cada canal y qué camino multipaso utilizar en cada envío de un mensaje. Esta es la política DRB ejecutada en cada encaminador DRB de la red. El siguiente punto explica cómo se realizan estas tareas en función de la carga presente en la red en cada momento.

4.7.2 Encaminamiento bajo DRB

Existen varias alternativas para hacer el dimensionamiento de los metacaminos y tomar la decisión sobre el uso de los caminos multipaso. Se pueden distinguir tres tipos de políticas dependiendo de si la determinación de las condiciones de tráfico y la configuración de los metacaminos se hace de manera "off-line" o en tiempo de ejecución. Podemos nombrar estas tres políticas como estáticas, semiestáticas o dinámicas.

En las políticas estáticas la determinación del tráfico y la configuración de los Supernodos se hace de manera "off-line". Las políticas semiestáticas determinan el tráfico de manera "off-line" y deciden una secuencia de configuraciones de Supernodos que serán aplicados secuencialmente en tiempo de ejecución siguiendo los patrones presentes en la aplicación. La tercera alternativa son las políticas dinámicas, las cuales determinan el patrón de tráfico de la aplicación mediante la monitorización de la actividad de las comunicaciones, y la decisión de la configuración de los Supernodos en tiempo de ejecución. A continuación se comentan las características y ventajas y desventajas de cada una de ellas:

4.7.2.1 Políticas estáticas

Para una aplicación dada, de la que se conocen los volúmenes de cómputo y de comunicación de cada una de las tareas que la componen, las políticas estáticas configuran unos metacaminos y una selección de camino multipaso a partir del análisis estático del código de la aplicación, es decir, sin ejecutarlo. La información que extraen de la aplicación son los requerimientos de comunicación y la asignación de canales a

enlaces de comunicación. Esta configuración se usa desde el principio de la ejecución y no se cambia nunca.

Esta aproximación estática, que no requeriría un gasto extra en tiempo de ejecución, tiene la desventaja de la dificultad intrínseca del análisis del código de la aplicación y de extraer las necesidades reales de comunicación de la aplicación a partir del código fuente, los datos u otra información proporcionada por el programador o el compilador. Estas políticas definen un índice y buscan una única solución que trata de optimizar ese índice. Ésta puede ser una buena solución media para toda la aplicación. Lo que realmente sucede es que, como los valores de los volúmenes de cómputo y comunicación no se pueden conocer de manera exacta y son sólo aproximados, podría hacer una asignación de metacaminos y caminos multipaso que no fuese adecuada. Incluso, la aplicación podría necesitar una pocas pero bastante diferentes configuraciones que cambian de una a otra a lo largo del tiempo. En este caso, la configuración seleccionada nunca se acoplaría totalmente a ningún patrón de la aplicación.

4.7.2.2 Políticas semiestáticas

La políticas semiestáticas tratan de superar las limitaciones de las políticas estáticas expuestas mas arriba. Estas políticas analizan la aplicación de manera "off-line" y deciden una serie de configuraciones que serán consecutivamente aplicadas en tiempo de ejecución. Con esta técnica se eliminan las desventajas de las políticas estáticas pero, ahora, todavía es más difícil extraer una secuencia de necesidades de comunicación de la aplicación y cuándo aparecerán esos patrones de tráfico. Además, necesitan un tiempo extra para cambiar las configuraciones en tiempo de ejecución.

4.7.2.3 Políticas dinámicas

Este tercer grupo de políticas determina el tráfico de la aplicación mediante la monitorización del estado de las comunicaciones y deciden que Supernodos y caminos multipaso utilizar en tiempo de ejecución. Estas políticas superan los problemas de las anteriores aproximaciones porque pueden adaptarse de manera dinámica al tráfico de comunicaciones en cada momento. Su desventaja principal, sin embargo, es el gasto en tiempo extra para monitorizar las condiciones de tráfico y para cambiar las configuraciones de los Supernodos, por lo que es importante limitar la perturbación que se genera en la operación de monitorización.

Aunque las políticas estáticas o semiestáticas pueden ser útiles en algunos casos, las políticas dinámicas son las que tienen un rango de validez más general y son las que

pueden producir un mejor resultado debido a sus características, siempre que el "overhead" que generen esté controlado. En este trabajo, nos hemos centrado, pues, en la definición de políticas dinámicas para DRB. Presentamos aquí una política que le llamamos encaminamiento DRB.

4.7.2.4 Encaminamiento DRB

El encaminamiento DRB se encarga de configurar dinámicamente los metacaminos y de distribuir los mensajes entre los caminos multipaso del metacamino con el objetivo de minimizar la latencia y utilizar uniformemente los recursos de la red de interconexión. Los fundamentos del encaminamiento DRB son:

- ✓ Detección de las condiciones del tráfico en la red de interconexión mediante la monitorización de la latencia experimentada por los mensajes en tránsito por la red.
- ✓ Configuración dinámica de los metacaminos dependiendo de esa latencia detectada.
- ✓ Distribución de los mensajes entre los caminos multipaso del metacamino.

Consecuentemente, el encaminamiento DRB se divide en tres fases:

- ✓ Fase 1: Monitorización de la carga de tráfico de la aplicación.
- ✓ Fase 2: Configuración de los metacaminos.
- ✓ Fase 3:Selección del camino multipaso.

Estas fases son unidades independientes entre ellas y son ejecutadas individualmente para cada canal de la aplicación en curso. Inicialmente, todos los canales se configuran mediante metacaminos canónicos, es decir, usando los caminos estáticos mínimos.

Con ayuda de la Figura 4-23, que muestra las acciones desarrolladas por el encaminamiento DRB al nivel de diagrama de flujo de procesos, vamos a describir la operatividad del algoritmo de encaminamiento DRB.

La actividad de monitorización de la latencia es llevada a cabo por los propios mensajes en tránsito por la red ("*Latency recording*") y su objetivo es registrar la latencia que el mensaje experimenta en su viaje hasta el destino. Cuando el mensaje llega a su destino con la información de latencia ("*Latency information*"), ésta se envía con un mensaje de reconocimiento ("*Ack message*")hacia atrás hasta el nodo origen del mensaje. La configuración de los metacaminos se realiza a nivel del canal cada vez que

una información de latencia llega al origen ("*Metapath Configuration*"). Esta latencia se usa para configurar un nuevo metacamino. La selección del camino multipaso también se realiza a nivel del canal cada vez que se inyecta un mensaje, tratando de elegir el camino multipaso de menor latencia ("*Multi Step Path Selection*").

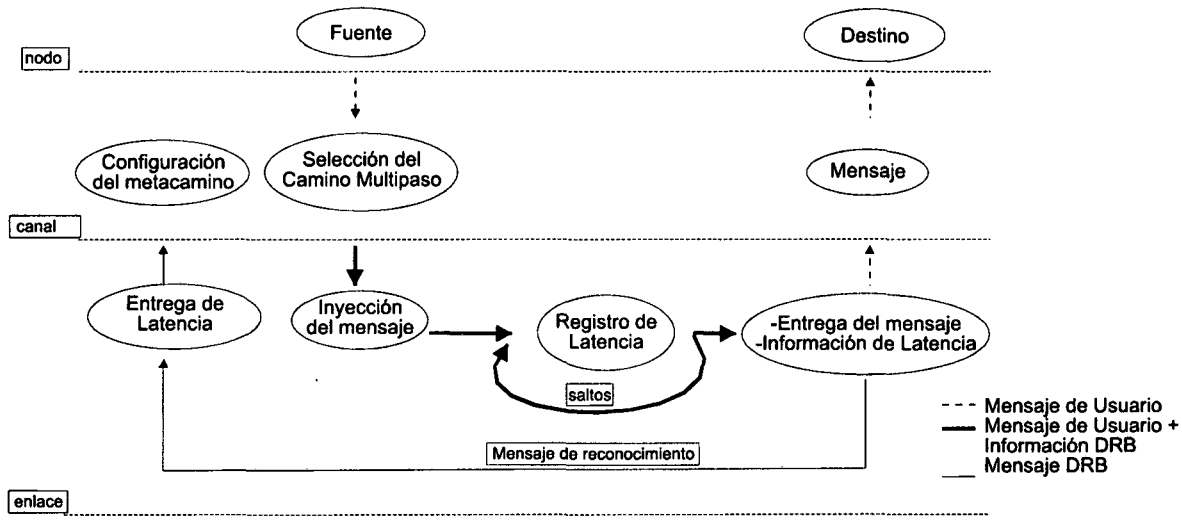


Figura 4-23 Encaminamiento DRB

La parte (a) de la Figura 4-24 muestra la situación inicial cuando los mensajes siguen el encaminamiento estático mínimo y la latencia se envía atrás hacia el origen. La parte (b) de la Figura 4-24 muestra la situación una vez que el metacamino se ha expandido y los mensajes se envían a través de varios caminos multipaso.

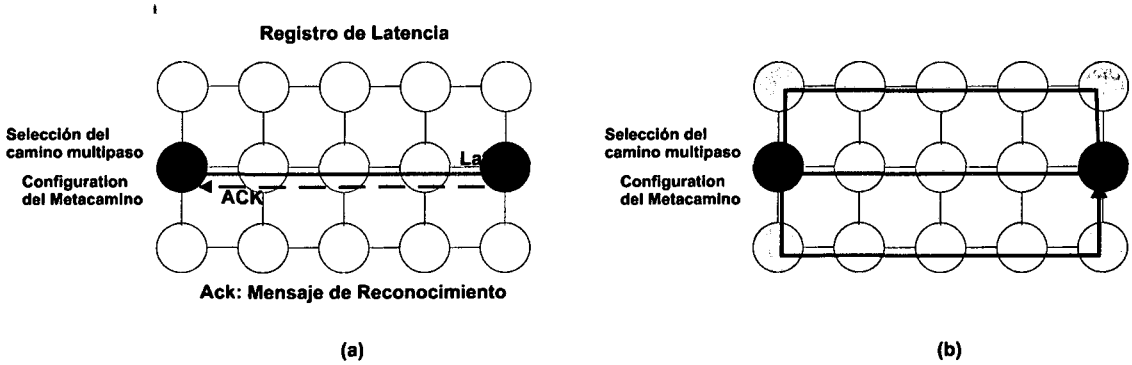


Figura 4-24 Fases del encaminamiento DRB

A continuación, se analiza cada una de las fases con detalle.

4.7.2.4.1 Fase 1: Monitorización de la carga de tráfico

La monitorización de la carga de tráfico es realizada sobre los propios mensajes por parte de los encaminadores DRB. En los mensajes se registra y transporta la información sobre la latencia sufrida durante su viaje hacia el destino. El mensaje

registra información sobre la contención que él experimenta en cada encaminador que atraviesa cuando se bloquea por causa de contención con otros mensajes.

Un mensaje, cuando es inyectado, lleva información en la cabecera de los destinos intermedios que debe visitar. La Figura 4-25 muestra el formato del paquete DRB. A la información a transmitir del usuario se le añade una cabecera formada por distintos campos. Esta cabecera se forma por la concatenación de los diferentes destinos a seguir. Este es el formato definido para la gestión de la función de encaminamiento por el encaminador DRB.

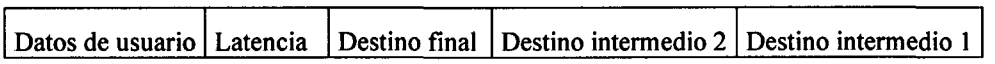


Figura 4-25 Paquete DRB

El protocolo para homogeneizar y simplificar la funcionalidad del encaminador DRB es el siguiente. El mensaje viaja entonces hacia el destino 1 siguiendo el encaminamiento estático mínimo como muestra la Figura 4-26 (a), cuando llega a él se elimina esa parte de la cabecera quedando como en la Figura 4-26 (b), y el mensaje continúa hacia el destino 2. Cuando llega a él, se repite la operación de eliminar la cabecera y se continúa hacia el destino final (Figura 4-26 (c)). El encaminador DRB siempre decide el siguiente enlace a utilizar en función de un destino que encuentra en el mismo lugar de la cabecera: al principio de todo.

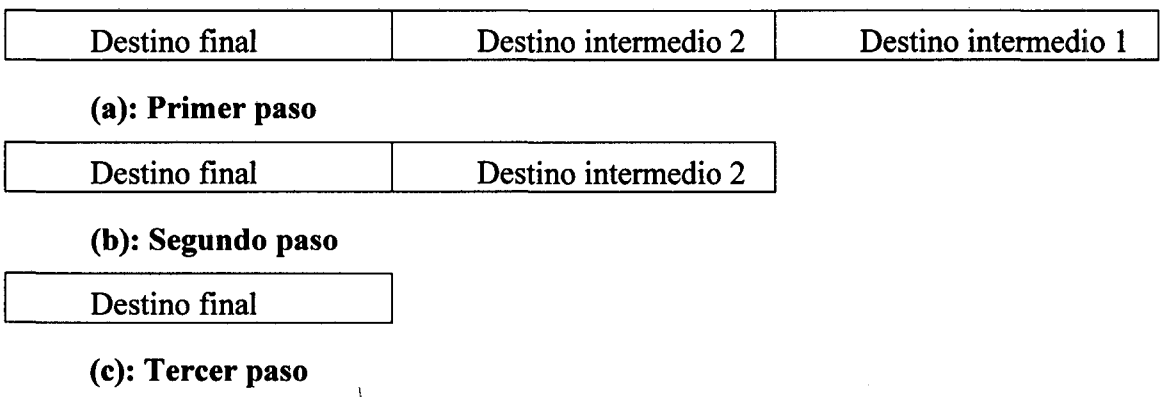


Figura 4-26 Cabeceras del mensaje DRB

La fase de monitorización determina la latencia del camino multipaso de acuerdo con la Ecuación 4-5.:

$$Latencia(MSP) = Tiempo\ de\ Transmisión + \sum_{\forall nodos \in MSP} Retardo de Encolamiento\ (Nodo)$$

Concretamente, cuando un mensaje llega a un encaminador y el siguiente enlace de salida que debe tomar esta siendo usado por otro mensaje, el primero debe ser encolado y esperar hasta que se libere el enlace que desea utilizar. El tiempo de permanencia en la cola es el que se registra y se va acumulando en un campo en la cabecera del mensaje.

Cuando un mensaje llega a su destino llevando información sobre la latencia que existe en el camino multipaso que atravesó, el encaminador DRB envía esta latencia hacia atrás de nuevo en un mensaje de reconocimiento (Figura 4-23). Este mensaje de reconocimiento debe tener la máxima prioridad en la red. De esta forma los nodos fuentes tienen información de la latencia, es decir, el grado de ocupación, de todos los caminos multipaso por los que están enviando mensajes.

Esta actividad de monitorización se solapa con el envío de los mensajes. Es una monitorización continua pero poco perturbadora y con un retraso igual a la latencia del mensaje del usuario más la latencia del mensaje de reconocimiento. Al ser esta actividad de monitorización llevada a cabo por todos los mensajes en la red de interconexión, su objetivo es conseguir identificar en cada instante el patrón local de tráfico y contención de su camino de comunicaciones presente en la red. En esa información local se incluye el efecto que tienen los otros mensajes en la latencia del mensaje considerado. Por tanto, se produce un efecto colectivo de influencias mutuas. La fase de monitorización se resume en el pseudocódigo de la Tabla 4-1.

Respecto del informe de la latencia sufrida por el mensaje, es decir, de la generación del mensaje de reconocimiento, existe otra alternativa a la presentada hasta aquí. En este aspecto, en lugar de informar la latencia al final del recorrido del mensaje cuando llega al nodo destino, se puede hacer antes. A esta segunda alternativa la llamaremos método de información de la latencia temprano. En este caso, se genera el mensaje de reconocimiento para informar de la latencia registrada cuando la misma supera un cierto valor umbral, que indica que se ha detectado un problema. Con esta opción, el algoritmo de configuración de caminos dispone de información actualizada en cuanto se produce el aumento de latencia. Esta alternativa, que ofrece mayores prestaciones, también requiere un mayor gasto de ancho de banda en enviar la información de monitorización, con lo que se debe buscar un compromiso. La Tabla 4-2 muestra el código modificado para la etapa de monitorización en el caso que se genera el mensaje de reconocimiento hacia el nodo origen en cuanto la latencia supera un cierto umbral.

Monitorización_de_la_carga_de_tráfico **(mensaje M, camino multipaso MSP)**

/*Realizado por el encaminador DRB*/

Inicio

1. Para cada paso del mensaje M,

1.1. Acumular la latencia (tiempo de espera en colas) para calcular la latencia(MSP) según Ecuación 4-5.

1.2. Si se llega a un destino intermedio, continuar hacia el próximo destino intermedio o hacia el destino final

2. Cuando el mensaje llega al destino final,

2.1. la latencia(MSP) se envía hacia atrás hasta el origen en un mensaje de reconocimiento

3. Cuando el mensaje de reconocimiento llega al origen,

3.1. latencia(MSP) se entrega a la función de Configuración_del_Metacamino(MSP, latencia(MSP))

Fin **Monitorización**

Tabla 4-1 Código de monitorización DRB

Estos dos aspectos han sido evaluados, tanto en su respuesta transitoria como en régimen estacionario, y sus conclusiones se presentan en el capítulo 6. Para ello el simulador funcional utilizado incorpora ambas versiones de la monitorización.

4.7.2.4.2 Fase 2: Configuración dinámica de los metacaminos

El objetivo de esta fase es determinar, para cada par fuente-destino, el tipo y tamaño del metacamino dependiendo de la latencia medida por los mensajes entre fuente y destino, mediante la configuración de unos Supernodos determinados a partir de sus parámetros de tipo y tamaño.

Monitorización_de_la_carga_de_tráfico2 **(mensaje M, latencia umbral L, camino multipaso MSP)**

/*Realizado en cada encaminador intermedio*/

Inicio

1. Para cada paso del mensaje M,

1.1. Acumular la latencia (tiempo de espera en colas) para calcular la latencia(MSP) según Ecuación 4-5.

1.2. Si se llega a un destino intermedio, continuar hacia el próximo destino intermedio o hacia el destino final

1.3. Si latencia(MSP) supera latencia umbral L, enviar mensaje de reconocimiento al origen

2. Cuando el mensaje llega al destino final,

2.1. La latencia(MSP) se envía hacia atrás hasta el origen en un mensaje de reconocimiento

3. Cuando el mensaje de reconocimiento llega al origen,

3.1. Latencia(MSP) se entrega a la función de Configuración_del_Metacamino(MSP, latencia(MSP))

Fin **Monitorización**

Tabla 4-2 Código de monitorización alternativa en DRB

Cuando un nodo fuente recibe una latencia de un camino multipaso, calcula la latencia del metacamino del que forma parte de ese MSP (usando la Ecuación 4-10: Latencia(P*)=$(\sum_{\forall s} Latencia(MSPs)^{-1})^{-1}$). Según esta latencia decide incrementar o reducir el tamaño de los Supernodos del metacamino dependiendo de si la latencia del metacamino está dentro o fuera de un intervalo definido por [*LatUmbral - Tol, LatUmbral + Tol*]. La latencia *LatUmbral* identifica el punto de saturación de la latencia (el punto de cambio de la parte plana de la curva a la parte de pendiente pronunciada) visto en la Figura 4-16 al principio de esta sección. El valor de tolerancia *Tol* define la desviación tolerada de la latencia para ese metacamino. El intervalo determinado por *LatUmbral* y *Tol* define el rango donde el metacamino no se cambia y

se acepta como válido. Este procedimiento hace que los canales se mantengan funcionando dentro de un nivel de carga que les proporciona una latencia baja y controlada alrededor de la *LatUmbral*. Si la latencia aumenta, se debe aumentar el metacamino para utilizar un mayor ancho de banda. Si la latencia disminuye y sale fuera del intervalo es porque ese canal dispone de un metacamino configurado demasiado grande y está usando unos recursos que debe liberar para otros canales. Por esto se debe disminuir su metacamino.

Con la latencia del metacamino, se calcula el ancho de banda actual del metacamino usando la Ecuación 4-11 (AnchodeBanda(P*)= Latencia(P*)$^{-1}$= ($\sum_{\forall s} AnchodeBanda(MSPs)$). Los tamaños de los Supernodos se modifican para dimensionar un nuevo metacamino de acuerdo con la relación entre el ancho de banda canónico del metacamino (*ABc*) y el ancho de banda actual (*AB*) Se pretende que el ancho de banda actual se sitúe entre un intervalo que nunca sea menor que el ancho de banda canónico, lo que implicaría que el canal no tendría ni el equivalente de un solo camino libre para enviar sus mensajes, ni mayor que un cierto número de veces ese ancho de banda canónico, lo que implicaría que el canal está usando "demasiados" recursos de la red en detrimento de otros canales.

Por lo tanto, se busca incrementar o decrementar el metacamino para que su ancho de banda este dentro de este intervalo definido por *Tol*. Suponiendo que cada camino multipaso añadido al metacamino aporta un ancho de banda equivalente al ancho de banda canónico, se busca en cuántos caminos se debe aumentar el metacamino para que su ancho de banda esté dentro del intervalo deseado. Esta actualización del metacamino se muestra en el procedimiento de la Tabla 4-3.

El valor de tolerancia *Tol* define la desviación tolerada del ancho de banda actual y el ancho de banda canónico del metacamino, *k* es un parámetro definible por el usuario que representa el uso que hace el canal del ancho de banda disponible en la red y se debe ajustar para maximizar su uso.

La configuración de los Supernodos aquí expuesta toma en consideración los valores de latencia en cada momento, así como las características topológicas de la red de interconexión y la distancia física entre fuente y destino, para balancear el ancho de banda que configura para el metacamino y su alargamiento correspondiente.

El pseudocódigo de la Tabla 4-4 muestra la fase de configuración del metacamino.

Compara *AB* con *ABc * Tol:*

✓ Si *AB < ABc * Tol*:

Incrementar la variación del tamaño *Δtamaño* de los Supernodos desde 1 hasta que:

$$ABc < AB + ABc * Tol * 1/k * \sum_{i=1}^{\Delta tamaño} i < ABc(1+1/k)$$

Ecuación 4-12 Incremento del Metacamino

e incrementar el tamaño del metacamino = tamaño actual del metacamino + *Δtamaño*;

✓ Si AB > ABc * Tol

Incrementar la variación del tamaño *Δtamaño* de los Supernodos desde 1 hasta que:

$$ABc < AB - ABc * Tol * 1/k * \sum_{i=1}^{\Delta tamaño} i < ABc(1+1/k)$$

Ecuación 4-13 Decremento del Metacamino

y decrementar el tamaño del metacamino = tamaño actual del metacamino - *Δtamaño*;

Tabla 4-3 Procedimiento de configuración de metacaminos en DRB

En el código de la Tabla 4-4, la implementación del paso 2 se realiza de la siguiente manera. Cuando llega un nuevo valor de latencia para un MSP, se resta de la latencia del metacamino el valor antiguo de latencia para ese MSP y se suma el nuevo recién llegado. Esto implica que el paso 2 se reduzca a una resta y una suma de números enteros. La implementación del paso 3 se realiza incrementando/decrementando una cantidad constante (*AB*) por múltiplos de otra cantidad constante (*ABc*Tol*1/k*) y comparándola con otro valor constante (*ABc ó ABc(1+1/k)*), lo cual son operaciones simples sobre números enteros.

Configuración_del_metacamino (Latencia umbral LatU, Tolerancia Tol);

/*Ejecutado en los nodos origen cada vez que llega una Latencia(MSP)*/

Variables Latencias_MSP: **Vector** [1..Número_de_MSP] **de entero**;

Inicio

1. Recibir una latencia(MSP);

2. Calcular la latencia(P*) usando Ecuación 4-10

$$\textbf{Latencia(P*)}=(\sum_{\forall s} Latencia(MSPs)^{-1})^{-1}$$

3. **Si** (Latencia(P*) > LatU+Tol) Incrementar el tamaño de los Supernodos según la Ecuación 4-12

while $ABc < AB + ABc * Tol * 1/k * \sum_{i=1}^{\Delta tamaño} i < ABc(1+1/k)$ **do**

Inc $\Delta tamaño$

Sino Si ((Latencia(P*) < LatU+Tol)) Decrementar el tamaño de los Supernodos según la Ecuación 4-13

while $ABc < AB - ABc * Tol * 1/k * \sum_{i=1}^{\Delta tamaño} i < ABc(1+1/k)$ **do**

Inc $\Delta tamaño$

SiFin

Fin Configuración_Del_Metacamino

Tabla 4-4 Código de configuración de metacaminos en DRB

4.7.2.4.3 Fase 3: Selección del camino multipaso

Esta fase se encarga de seleccionar un camino multipaso para cada mensaje para distribuir equilibradamente la carga de comunicaciones entre los caminos multipaso de un metacamino. Para cada mensaje que se envía, se selecciona un MSP dependiendo de sus anchos de banda en una relación donde el MSP de mayor ancho de banda es el más frecuentemente utilizado. Así pues, los mensajes se distribuyen entre los caminos multipaso en proporción al ancho de banda de cada uno de ellos. De esta forma, la carga

se distribuye entre todos los caminos multipaso del metacamino, pero de manera que los que tienen mayor capacidad disponible reciben mayor número de mensajes. Es decir, un MSP que tenga un ancho de banda doble que otro, recibirá también el doble de mensajes que el de menor ancho de banda. De esta manera, se distribuye la carga entre todos los MSPs del metacamino, lo que permite balancear la carga entre ellos y mantener monitorizados todos los MSPs.

Supongamos que MSP(k) es el k-ésimo camino multipaso del metacamino y que AB(MSP(k)) es su ancho de banda asociado. Entonces, se usan los anchos de banda para ordenar los MSP mediante el método de construir una función probabilística de distribución acumulativa discreta. Para ello, se toma el ancho de banda de cada MSP como valor de una distribución de probabilidad discreta de los caminos multipaso. El procedimiento de selección del camino multipaso se muestra a continuación:

- ✓ Primero, convertir los valores de la distribución en una función de distribución acumulativa Π, obteniendo las proporciones Π[MSP(k)] = probabilidad[MSP(k)<k], sumando y normalizando los anchos de banda discretos de cada MSP del siguiente modo:

$$\Pi[MSP(k)] = \frac{\sum_{l=1}^{k} \text{AB(MSP}(l))}{\sum_{l=1}^{s} \text{AB(MSP}(l))}; \; P(MSP(s)) = 1; \; (s = Numero\ de\ MSPs)$$

Ecuación 4-14 Función acumulativa de MSPs

- ✓ Segundo, generar un valor aleatorio R entre [0,1)

- ✓ Tercero, encontrar los k-ésimo y (k-1)-ésimo MSPs tal que

$$\Pi \text{ [MSP(k-1)<k-1]} < \text{R} <= \Pi \text{ [MSP(k)<k]}$$

Ecuación 4-15 Selección de un MSP

- ✓ Cuarto, seleccionar el k-ésimo camino multipaso para enviar el mensaje.

El pseudocódigo de la Tabla 4-5 muestra la fase de selección del camino multipaso. Mostramos a continuación con un ejemplo numérico la implementación real que pone en práctica las fórmulas anteriores. Supongamos que tenemos 5 caminos multipaso pertenecientes al mismo metacamino y cuyos anchos de banda son:

MSP(1) = 5; MSP(2) =8; MSP(3) =3; MSP(4) =4; MSP(5) =9

Selección del camino multipaso()

/*Ejecutado en el nodo fuente cada vez que se inyecta un mensaje/*

Inicio

1. Construir la función acumulativa de distribuciones sumando y normalizando los anchos de banda de los caminos multipaso según la Ecuación 4-14

2. Generar un número aleatorio entre [0,1)

3. Seleccionar un MSP usando la función de distribución acumulativa formada según la Ecuación 4-15

4. Inyectar el mensaje en la red

4.1. Construir una cabecera múltiple con los destinos intermedios que forman el MSP y el destino final

4.2. Concatenar los datos con la cabecera

4.3. Inyectar el mensaje con formato DRB

Fin Selección_del_Camino_Multipaso

Tabla 4-5 Código de selección de MSPs en DRB

Entonces, formamos la función de distribución de probabilidades acumulada y normalizada de la siguiente forma:

Π[MSP(1)]=5/29; Π[MSP(2)]=13/29; Π[MSP(3)]=16/29; Π[MSP(4)]=20/29;

Π[MSP(5)]=29/29

Estas probabilidades pueden verse, antes de normalizar, gráficamente como:

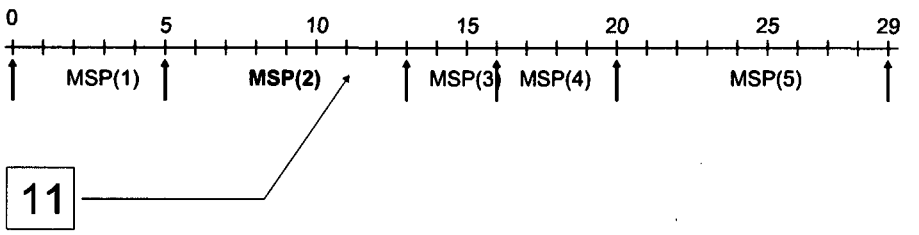


Figura 4-27 Distribución de probabilidades de los MSPs

En la implementación real, se opera sin normalizar para evitar la división y la operación con números en punto flotante. Por lo tanto, a continuación, se genera un número entre 0 y 29 y se elige el MSP cuyo rango de probabilidades comprende al número generado, tal y como se muestra en la figura anterior, donde suponiendo que el número generado aleatoriamente fuese el 11, se elegiría el MSP2. De esta manera, las operaciones implicadas son la generación aleatoria de un número entero y la búsqueda por comparación entre los anchos de banda de los MSPs del rango de valores en que se encuentra el número generado.

4.7.2.4.4 Análisis de DRB

Hasta aquí se ha desarrollado la explicación de cómo se encaminan los mensajes usando DRB. Es importante remarcar que, para conseguir una distribución uniforme de la carga de comunicaciones de manera efectiva, se necesita una acción global y que, por esta razón, todos los pares fuente-destino de la aplicación están capacitados para expandir sus caminos dependiendo de la carga de tráfico entre ellos durante la ejecución del programa.

En este sentido, la fase de monitorización realizada para cada canal, como se ha visto, pretende conocer el estado de la red en la zona que esta utilizando para enviar los mensajes. Como todos los canales realizan las mismas acciones, si una zona empieza a cargarse y su latencia supera los valores tolerados, los canales decidirán "apartarse" de ella hacia otras zonas.

En el caso del balanceo de la carga de cómputo, puede considerarse migrar tareas desde cualquier procesador a cualquier otro en la red. Pero en el caso que nos ocupa de las comunicaciones, la búsqueda de nuevas zonas tiene sólo sentido hacerla hacia zonas contiguas a las que se están usando, pues, los nodos origen y destino no cambian, y al final los mensajes parten de uno para ir hasta el otro. Con esta acción de búsqueda de nuevos caminos, los canales pueden "invadir" el espacio que estaban usando otros canales afectándoles a estos últimos. Entonces, estos últimos también buscarán nuevas zonas menos congestionadas de la red. De esta manera, se consigue un efecto global de balancear la carga de comunicaciones desde las zonas más cargadas hacia las menos cargadas.

El método DRB aprovecha la localidad espacial y temporal de las comunicaciones del programa paralelo, como los sistemas de memoria "cache" lo hacen de las referencias a memoria. El algoritmo adapta las configuraciones de los Metacaminos al patrón de tráfico de la aplicación en cada instante. Mientras este patrón se mantiene, la latencia se mantiene baja y la función de configuración de los metacaminos no se activa.

Cuando la aplicación cambia para crear un nuevo patrón de comunicaciones y la latencia de los mensajes cambia, la monitorización de DRB detecta esos cambios y se activa la configuración de unos nuevos parámetros para los metacaminos para adaptarse a la nueva situación.

Cómo todos los métodos que se basan en el comportamiento pasado para pronosticar cuál será la evolución en el futuro, DRB es útil para patrones de comunicación persistentes, que son los que pueden causar las peores situaciones de saturación (dado su persistencia, precisamente) Asimismo, DRB reduce la latencia de inyección de mensajes en la red mediante la configuración de diversos caminos disjuntos entre cada par fuente y destino, permitiendo enviar mensajes por todos los enlaces del nodo origen simultáneamente. Además, la adaptación de los metacaminos es específica y puede ser diferente, si se desea, para cada canal de la aplicación dependiendo de su distancia física o de la latencia que sufre. En este sentido, DRB se puede adaptar al comportamiento de diferentes patrones de comunicación.

El efecto perseguido con el método DRB es permitir un nivel más alto de tráfico aceptado, es decir, lograr que la saturación de la red se produzca a tasas más altas de tráfico de la red.

Con la capacidad de DRB de adaptarse a picos de tráfico de la aplicación, se consigue que la granularidad de los procesos de la aplicación (la relación entre los volúmenes de cómputo y comunicación) pueda ser menor, es decir, más comunicaciones por menos cómputo. Además, esta granularidad puede presentar mayores variaciones entre las diferentes tareas de la aplicación o a lo largo del tiempo, porque con DRB estas variaciones se toleran mejor. Estas dos cuestiones sobre la granularidad son de una gran importancia para el programador-usuario del computador de altas prestaciones en el sentido que facilitan la creación de programas paralelos.

Asimismo, la distribución de tareas entre nodos de cómputo se simplifica porque con DRB se pretende que todas las latencias reales se mantengan próximas a las latencias en ausencia de colisiones y, por lo tanto, el retardo de comunicaciones se puede conocer anticipadamente.

Dado este escenario, la única cuestión de la cual el programador debería preocuparse es que los requerimientos de ancho de banda totales en la aplicación no excedan en ningún momento el ancho de banda de la red: la distribución de ese ancho de banda ya no es más una preocupación porque DRB se encarga de ella.

El método DRB es independiente de la topología y de la técnica de control del flujo y, de hecho, se puede aplicar a cualquier red, regular o irregular, con cualquier control del flujo ("*Store-and-forward*", "*Wormhole*" o "*Cut-through*").

4.7.2.4.5 Estudio de anomalías de comunicación: "deadlock", "livelock", "starvation" y orden de los mensajes.

Dependiendo de la topología, DRB puede introducir la posibilidad de aparición de "*deadlock*" en la red de interconexión. Por lo tanto, debe usarse alguna de las técnicas presentadas en el capítulo 2 de evitación o detección y recuperación de "*deadlock*". Por ejemplo, puede usarse la metodología presentada por J.Duato en [41] consistente en la utilización de canales virtuales y la provisión de canales de escape en caso de "*deadlock*". En este caso, como los Caminos Multipaso que siguen los mensajes bajo DRB están compuestos de varios encaminamientos estáticos simples, DOR por ejemplo, las técnicas que eliminan "*deadlock*" para el encaminamiento estático son válidas para DRB, simplemente aplicándolas en cada uno de los pasos del Camino Multipaso, siempre que el mensaje se almacene completamente en el nodo en cada paso intermedio.

En la implementación que hemos realizado de DRB para experimentar y evaluarlo mediante simulación, hemos utilizado una técnica de detección y recuperación de "*deadlock*", consistente en la detección de ciclos de mensajes y la extracción de un mensaje de la red para romper el ciclo.

Respecto al problema de "*livelock*", se puede observar que DRB no lo presenta ya que por propia definición nunca configura caminos de longitud infinita y los mensajes siempre llegan a su destino en un número de pasos determinado. Tampoco presenta problemas de "*starvation*" porque, por un lado, no se le impide a ningún nodo que inyecte mensajes durante un tiempo indefinido, y, por el otro, no es posible que un mensaje quede indefinidamente bloqueado en un encaminador intermedio del camino porque todos los mensajes tienen oportunidad de acceder a los enlaces de salida.

Finalmente, DRB puede producir el efecto de desordenar los mensajes. Pero solo se deben ordenar los mensajes pertenecientes al mismo canal lógico, es decir, un par fuente-destino. Este ordenamiento es fácil de conseguir simplemente numerando los mensajes al inyectarlos en el nodo fuente y ordenándolos según ese índice en el nodo destino. Adicionalmente, se puede utilizar la técnica de "*message pre-fetching*" para ocultar la desordenación de los mensajes. Esta técnica solicita la recepción de los mensajes un tiempo antes de que se necesiten para que estén disponibles en cuanto sean necesarios.

4.7.2.4.6 Comparación con otros métodos

Muchos de los métodos adaptivos intentan modificar los caminos a utilizar cuando un mensaje llega a un nodo congestionado. Este es el caso, por ejemplo, del "Chaos Routing" [17] introducido en el capítulo 2. Este método introduce aleatorización para desencaminar mensajes cuando éstos se bloquean. La diferencia con DRB es que DRB no actúa al nivel del mensaje individual, sino que intenta adaptar todo el flujo de comunicación entre los nodos fuente y destino hacia caminos no congestionados.

En este sentido, DRB puede entenderse como un método de los llamados "encaminamiento desde la fuente - *source-routing*" en los que la determinación del camino se hace desde el nodo fuente antes de inyectar el mensaje. Esta técnica contrasta con la de otros métodos adaptivos en los que las decisiones de encaminamiento se realizan nodo a nodo dependiendo de las condiciones encontradas, sean locales o con cierta información global.

Los métodos de encaminamiento aleatorio introducidos por Valiant [130] y May [88], y ya comentados en este trabajo, distribuyen los requerimientos de ancho de banda uniformemente sobre el sistema completo, pero a expensas de doblar la longitud de los caminos en media. Un análisis mas detallado muestra que los caminos de longitud máxima en la red no son alargados, mientras que los caminos de longitud uno, en media, se alargan hasta la distancia media de la red para redes regulares. Esto demuestra que se perjudican excesivamente los caminos cortos. Esto es debido a que dicho método es un método "ciego" que no tiene en cuenta el tráfico presente en la red y distribuye los mensajes "a fuerza bruta" sobre la red entera.

Aunque DRB comparte ciertos objetivos con este tipo de encaminamiento aleatorio, la principal diferencia es que DRB sí tiene en cuenta factores como el tráfico o la distancia entre fuente y destino. Es por ello que DRB intenta no solo mantener una alta tasa de mensajes entregados sino también mantener controlada la latencia de los mensajes individuales porque el alargamiento de los caminos se puede controlar. Esto es contrario al encaminamiento aleatorio que, de media, duplica la longitud de los caminos que recorren los mensajes con el negativo efecto mencionado sobre los caminos más cortos.

El método DRB es una generalización del encaminamiento aleatorio y puede verse como un superconjunto de los métodos estático y aleatorio. El encaminamiento estático es un caso extremo de DRB en el que los dos supernodos, fuente y destino, contienen solo un nodo, el fuente o el destino, respectivamente. El encaminamiento

aleatorio es el otro extremo en el cual el supernodo fuente contiene todos los nodos de la red y el supernodo destino solo contiene el nodo destino.

Ideas de distribuir el encaminamiento parecidas a DRB han sido utilizadas en sistemas reales, como el SP2 de IBM o el CS-2 de Meiko [11]. El algoritmo de encaminamiento del supercomputador SP2 de IBM ofrece una solución de distribución de caminos similar a DRB pero más restrictiva, menos flexible y no adaptiva en tiempo de ejecución. Esta solución se llama "*Route Table Generator*" (RTG) [122], la cual selecciona estáticamente cuatro caminos para cada par fuente destino los cuales son utilizados en una política "*roud-robin*" con objeto de utilizar la red uniformemente. El computador Meiko CS-2 también preestablece todos los caminos entre fuente y destino y selecciona cuatro caminos alternativos para balancear el tráfico de la red [16].

El siguiente punto presenta un ejemplo básico de utilización de DRB donde se muestra el efecto conseguido sobre las latencias de los mensajes.

4.7.3 Ejemplo básico

Después de completar la presentación teórica de DRB en los puntos anteriores, en este punto se muestra un ejemplo sencillo pero completo de cómo opera DRB. A través de una configuración de canales simple, se muestra la metodología de trabajo con Supernodos y metacaminos y se muestran los resultados obtenidos. Este ejemplo tiene como objetivo no mostrar todas las posibilidades y consecuencias de DRB, sino solamente los principios básicos de funcionamiento de DRB, por eso es un ejemplo donde se prima más la claridad que la profundidad.

Para cumplir esos objetivos, se considera una red de interconexión pequeña con unos pocos canales definidos expresamente de manera que haya unos canales que colisionan formando un "*hot-spot*", mientras que otros canales disponen de manera exclusiva del camino que utilizan y, por lo tanto no sufren el "*hot-spot*". Se evalúa esta configuración de canales con el modelo analítico presentado en el capítulo anterior y se muestra el comportamiento en latencia de este caso sin aplicar DRB. La elección del modelo analítico frente al funcional en este caso se debe a que sólo interesa el resultado estacionario y con el simulador analítico obtenemos los mismos resultados que con el funcional pero en menor tiempo.

A continuación, se diseñan unos Supernodos que configuran unos metacaminos para los canales que sufren grandes retardos por colisionar en el "*hot-spot*". Con esta configuración los canales disponen de mas caminos para enviar sus mensajes y por lo tanto de un mayor ancho de banda. Precisamente, los nuevos caminos que usan los

canales ocupan el camino de los canales que circulaban aislados, con lo que se empeora su situación. Se vuelve a evaluar la red con el simulador analítico y se ven los resultados de latencia.

El ejemplo mostrado dispone de seis canales sobre una topología toro como muestra la Figura 4-28. Existen cuatro canales (C1, C2, C3 y C4) que comparten su camino a lo largo de una serie de enlaces y otros dos canales (C5 y C6) que con independientes. La Tabla 4-6 muestra el conjunto de fuente y destino para cada canal.

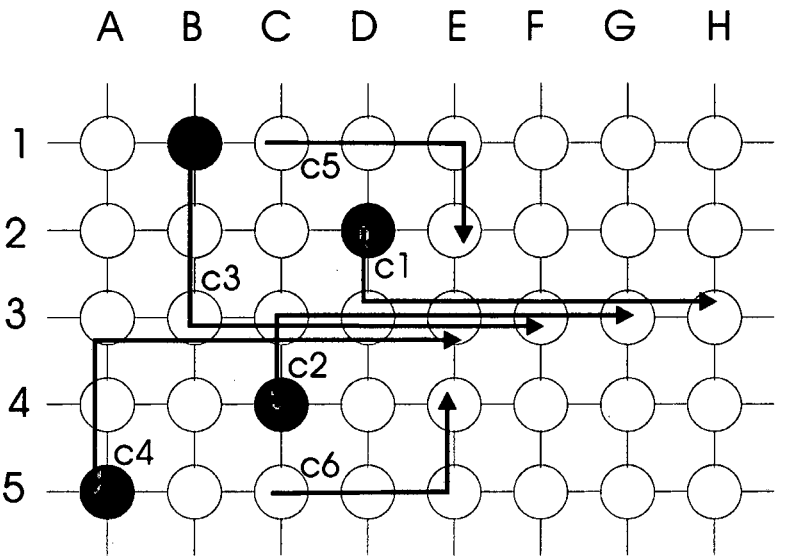


Figura 4-28 Ejemplo básico sin DRB

Canal	C1	C2	C3	C4	C5	C6
Fuente	D2	C4	B1	A5	C1	C5
Destino	H3	G3	F3	E3	E2	E4

Tabla 4-6 Conjunto de canales del ejemplo básico

La Figura 4-29 muestra el resultado en latencia para los canales C1, C2, C3 y C4 en este caso cuando se evalúa para unos intervalos de generación de mensajes en el rango 10-1000 ciclos entre mensajes (de 10, 25, 50, 100, 150, 200, 250, 500, 1000) sin usar el método DRB. El tiempo de transmisión para cada canal es de 25 ciclos para todos ellos. La latencia para C5 y C6 es de 25 ciclos, es decir, el tiempo de transmisión puro, pues no colisionan con nadie. La gráfica por tanto se muestra a partir de este tiempo de transmisión, que es la latencia subyacente mínima sin colisiones. Se observa en la curva de la gráfica que se produce un crecimiento brusco de la latencia cuando el intervalo de la carga disminuye por debajo de 100 ciclos. Este "*hot-spot*" se produce

porque existe una incorrecta distribución de los mensajes en la red de interconexión, ya que ninguno de los canales C1, C2, C3 o C4 comparten sus nodos fuente o destino.

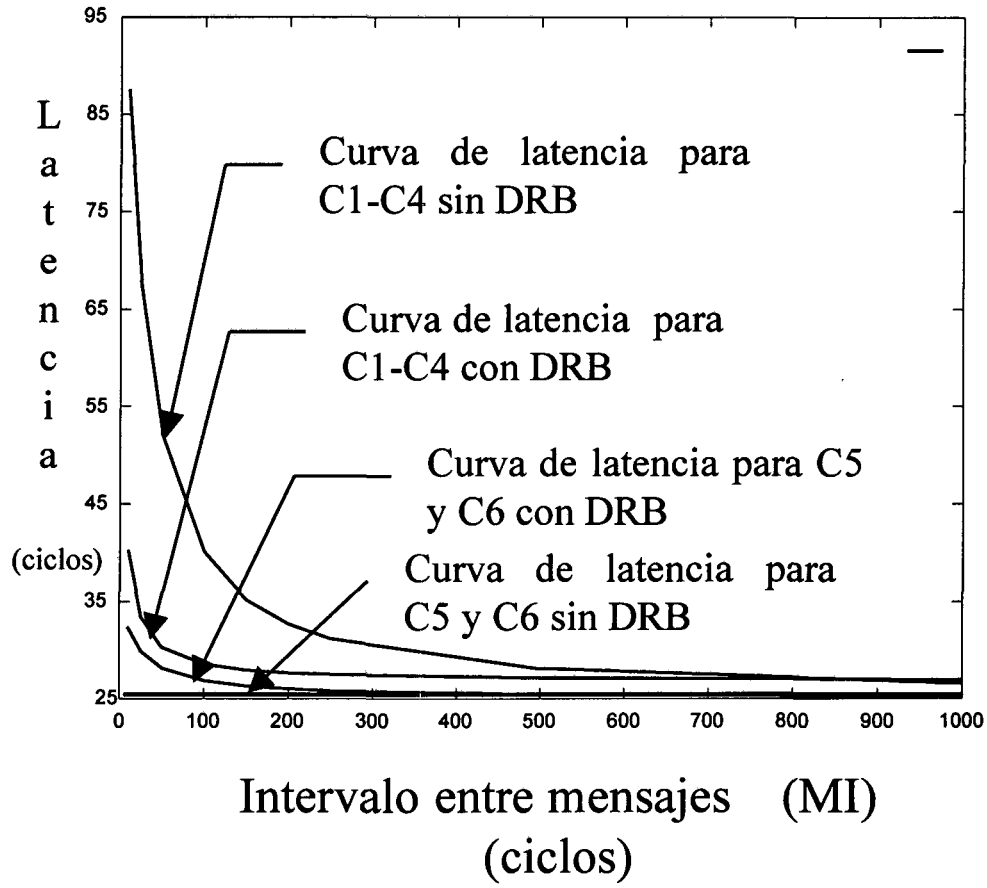


Figura 4-29 Curvas de latencia para el ejemplo básico

A continuación, configuramos unos Supernodos para cada uno de los canales problemáticos consistentes en la fila de nodos que lo contienen. La

Tabla 4-7 muestra la configuración de caminos alternativos establecida por DRB de forma dinámica mediante la monitorización, creación del Supernodo y los caminos multipaso, y la distribución por los diferentes caminos multipaso. Por simplicidad, el supernodo destino en este caso se ha reducido al nodo destino principal, con lo cual los caminos multipaso se componen de uno o dos pasos como máximo. La configuración aplicada busca nuevos caminos cercanos al camino original que se supone que están libres o menos ocupados. La Figura 4-30 muestra la configuración de nuevos caminos disponibles con el método DRB activado.

La Figura 4-29 muestra las nuevas configuraciones de latencia en la zona de canales para los canales C1, C2, C3 y C4. La latencia, en la zona de carga de saturación (Intervalo entre mensajes < 100 ciclos), es entre cuatro y cinco veces menos que cuando no se usa DRB. Además, se observa que ahora los canales trabajan la mayor parte de su tiempo en una zona de latencia plana, siendo ésta uniforme ya que presenta una varianza mucho menor que en el caso en el que no se usa DRB.

Es importante remarcar aquí que se ha reducido la latencia para todos los valores de carga (intervalo entre mensajes), de forma especialmente significativa para los valores de carga elevada (MI < 200-300) y que el comienzo de la zona de saturación (la pendiente elevada en la gráfica) se ha desplazado hacia la región de carga elevada (sin DRB estaba en un MI=150-100 y con DRB está en MI=50-30). De ahí que ahora la zona plana de la curva (latencia constante para cambios de MI) ha aumentado, con lo que aumenta el rango de uso de la red en la zona de comportamiento de latencia constante.

Canal	**Fuente**	**MSPs**	**Destino**
C1	D2	D2-H3, D2-A2-H3, D2-H2-H3	H3
C2	C4	C4-G3, C4-G4-G3, C4-H4-G3	G3
C3	B1	B1-F3, B1-F1-F3, B1-G1-F3	F3
C4	A5	A5-E3, A5-E5-E3, A5-F5-E3	E3
C5	C1	C1-E2	E2
C6	C5	C5-E4	E4

Tabla 4-7 Configuración de caminos multipaso del ejemplo básico

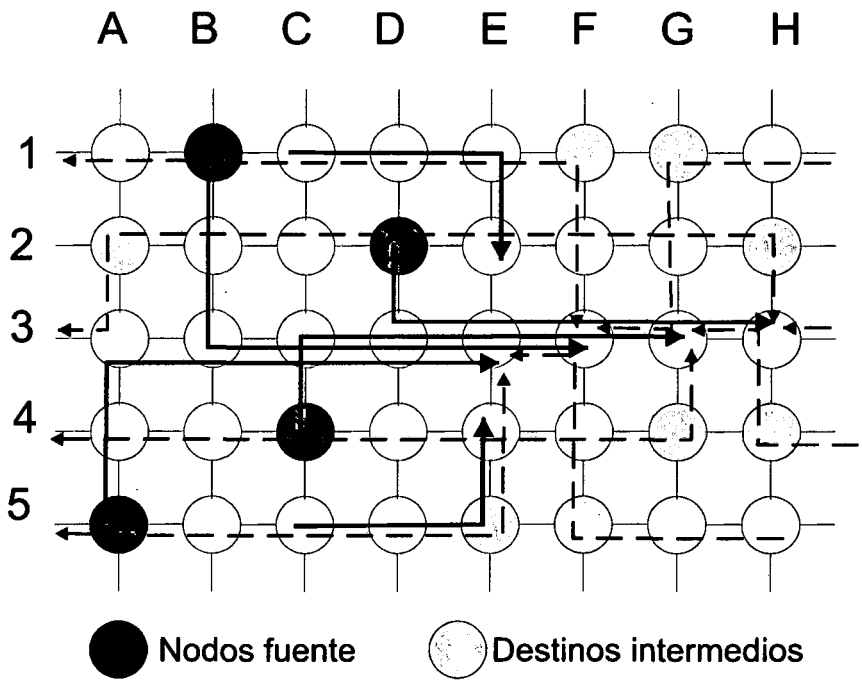


Figura 4-30 Ejemplo básico con DRB

Además, se observa el efecto dinámico de la apertura de los canales a mas caminos alternativos, según el encaminamiento DRB descrito anteriormente. Cuando la carga es baja, la latencia también es baja y la fase de configuración de metacaminos no se activa y los mensajes siguen por los caminos mínimos estáticos. Esta cuestión se observa en las zonas de baja carga de la Figura 4-29 en que las latencias en el caso de aplicar DRB y no aplicarlo coinciden, lo que significa que no se usan caminos alternativos. Cuando la latencia empieza a incrementarse y supera un cierto valor umbral, se abren los caminos y entonces las mejoras en reducción de la latencia son muy importantes.

También es importante observar el resultado para los canales independientes C5 y C6. Como ahora deben compartir su camino con los mensajes de otros canales, se ven afectados negativamente. Observamos en la Figura 4-29 que su incremento en latencia es muy pequeño por lo que se ven muy poco afectados por la distribución de caminos aplicada a C1, C2, C3, y C4.

Con este ejemplo, se observa el balanceo efectivo de la carga de comunicaciones conseguido por DRB a través de la expansión de los caminos mediante la configuración de supernodos y metacaminos y la selección de caminos multipaso para el envío de mensajes. El resultado obtenido es que al distribuir las comunicaciones hemos conseguido aumentar el número de caminos que trabajan en la zona plana de la curva, lo

cual garantiza un intervalo mayor de carga para el cual la latencia se mantiene casi constante y en valores bajos.

4.7.4 Implementación de DRB. Encaminador DRB

En este punto, se aborda la implementación física del método DRB y de los aspectos de diseño que debería incluir un encaminador para soportar las funciones DRB. El método DRB hasta ahora descrito se basa en las tres fases de monitorización, configuración de los metacaminos y selección de los caminos multipaso. La implementación de estas tres funciones implementa el método DRB en una red de interconexión. Esta implementación requiere modificaciones tanto en las funciones de los encaminadores como en la estructura de los paquetes para soportar dichas fases. Las tres funciones del encaminamiento DRB, monitorización, configuración de los metacaminos y selección de los caminos multipaso, se implementan en diferentes puntos del sistema de comunicaciones.

La función de monitorización, encargada de registrar la latencia de los mensajes durante su viaje y enviarla al nodo origen, se implementa en los encaminadores, al nivel de la capa de transporte. Las otras dos funciones, configuración de los metacaminos y selección de caminos multipaso, no necesitan implementarse en los encaminadores, lo cual no complica el diseño del encaminador DRB, sino que son funciones que se implementan en las interfaces de red (NI: "*Network Interface*") de cada nodo que accede a la red de interconexión.

Antes de describir la implementación de cada fase, vamos a comenzar el estudio de la implementación física de DRB por la modificación de la estructura del paquete necesaria para soportar las funciones de DRB. Las modificaciones necesarias son dos: la primera, para implementar los caminos multipaso, y la segunda, para almacenar la latencia sufrida.

Para el primer cometido, el paquete debe estar compuesto de una cabecera multidestino. La cabecera multidestino se implementa concatenando varios destinos consecutivamente en la cabecera como se muestra en la Figura 4-31. Además, existe un bit que indica si existen más cabeceras o no. De esta manera, el encaminador DRB, encamina los mensajes que todavía no han llegado a su destino como cualquier encaminador convencional, siguiendo encaminamiento mínimo estático. Cuando la cabecera de un mensaje indica que ha llegado a su destino, intermedio o final, porque su cabecera coincide con el nodo en curso, el encaminador DRB consulta el bit correspondiente. En el caso de que indique que existen mas destinos, elimina una

cabecera y sigue encaminando el mensaje hacia el nuevo destino. En el caso de que sea el destino final, encola el paquete en la cola de entrega de paquetes al nodo.

Además, el paquete lleva información de cuál es el MSP por el que ha viajado (es decir, los destinos intermedios) y el nodo origen. Esta información se incluye en el mensaje de reconocimiento junto con la latencia sufrida en ese MSP para ser enviado al nodo fuente.

Para la segunda función, el paquete reserva un espacio para guardar la latencia por él sufrida en un campo específico. Este campo es de tamaño adecuado para guardar un número entero de rango suficiente (Figura 4-31).

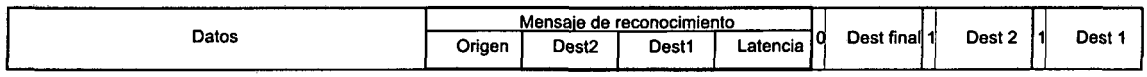


Figura 4-31 Formato del paquete DRB

Con este formato de paquete, cuando un mensaje está siendo encaminado por la red de interconexión, sigue un encaminamiento mínimo determinado por la primera cabecera, por ejemplo, por dimensiones DOR, el cuál es transparente a la existencia del mecanismo DRB, ya que las demás cabeceras se tratan como puro mensaje. Cuando un paquete llega a destino, sólo debe detectarse si es un destino intermedio o no. En el caso de que sea destino intermedio, se elimina una cabecera del paquete y se realiza la función de encaminamiento con la siguiente cabecera. En el caso de que sea el destino final, el paquete se encola en la cola de consumo del encaminador correspondiente. Esta funcionalidad la realiza el módulo de *Detección y Eliminación de Cabeceras Intermedias* (DEC) que está dentro del módulo de encaminamiento y arbitraje mostrado en la Figura 4-32.

Por lo tanto, los encaminadores de la red de interconexión sólo deben ser modificados, respecto a su estructura básica presentada en el capítulo 2, en que el módulo de encaminamiento y arbitraje debe ser capaz de detectar y eliminar cabeceras intermedias.

Para la función de registro de la latencia debe hacerse lo siguiente. Cuando un paquete se bloquea en un encaminador intermedio porque el enlace de salida no está disponible, se debe acumular el tiempo de espera y registrarlo en el campo de latencia del paquete. Este tiempo se puede contar en ciclos de reloj del encaminador o cualquier unidad superior que sea útil para informar de la latencia. En todo caso, es una medida local que solo afecta al encaminador en el que está el paquete retenido. Esta medida se puede hacer tomando el tiempo del reloj cuando el paquete se encola, volviéndolo a

tomar cuando prosigue el camino y restando ambos valores y sumándolo a la latencia acumulada hasta ese momento.

Esta operación puede implementarse de la siguiente forma. Supongamos que un paquete llega a un nodo con un cierto valor de latencia acumulado *L1* y que este paquete debe encolarse en el instante que el valor del "*timer*" interno es *T1* y que continúa su viaje por la red en tiempo *T2*. Entonces, el nuevo valor de latencia a almacenar en el paquete debe ser *L1*+(*T2*-*T1*), que es lo mismo que *L1*-*T1*+*T2*. Por lo tanto, la operación se implementa mediante la resta de *L1* y *T1* cuando el paquete se encola y la suma de *T2* cuando el paquete reanuda su trayecto. Con esta manera de cálculo no es necesario almacenar el valor temporal de *T1* en un registro específico. Por lo tanto, el módulo *AL* de la Figura 4-32 se compone de un complementador y un sumador.

Como el número de paquetes que se encolan es conocido y finito para un encaminador, y frecuentemente pequeño, se puede configurar el espacio y recursos necesarios para estas operaciones. La Figura 4-32 muestra las modificaciones necesarias en el encaminador básico para implementar el encaminador DRB.

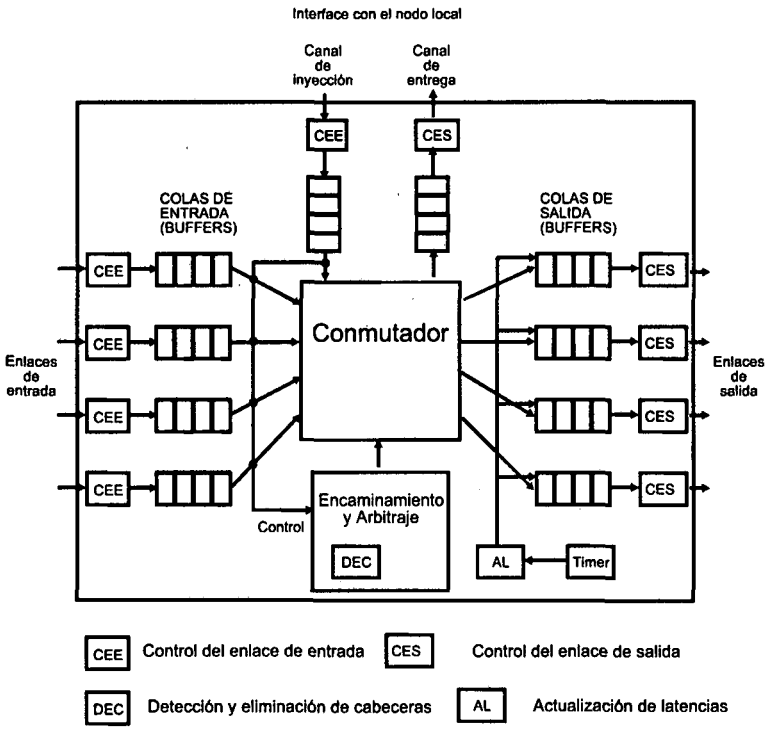


Figura 4-32 Estructura del encaminador DRB

Para realizar la función de arbitraje y configuración del "*crossbar*" interno en el módulo de encaminamiento y arbitraje, en el que se deben administrar peticiones de *n* enlaces de entrada con *m* enlaces de salida, se puede utilizar el sistema de arbitraje modular diseñado en [86] para sistemas multiprocesadores de múltiples "buses". Sus

funciones son seleccionar el enlace de entrada ante peticiones simultáneas por un mismo enlace de salida y la asignación de los elementos del "*crossbar*" para realizar la conexión. El sistema descrito en [86] es regular, modular, de bajo costo, rápido y fácilmente expansible. Además, puede funcionar de manera síncrona o asíncrona y puede manejar cualquier tamaño de "*crossbar*".

Finalmente, cuando un paquete, llega a su nodo destino y se entrega al proceso correspondiente, el nodo destino se encarga de generar un mensaje de reconocimiento hacia el origen con la información de latencia y del camino multipaso donde se produjo esa latencia (Figura 4-34). Este es un mensaje de control muy corto que tiene el formato especificado en la Figura 4-33. Los campos que aparecen son el destino donde se debe enviar el mensaje, que es el *origen* del que partió en mensaje de datos, la *latencia* sufrida y el camino multipaso al que pertenece la latencia, identificado por sus destinos intermedios *Dest1* y *Dest2*. Este mensaje debe ser inyectado inmediatamente en la red en cuanto se produce la recepción del mensaje original.

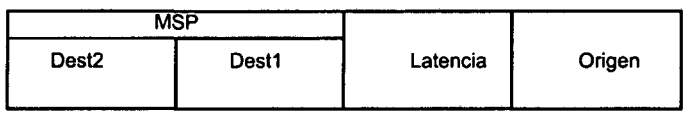


Figura 4-33 Formato del mensaje de reconocimiento

En el caso de utilizar un sistema de informe de la latencia temprano en el que en el mensaje de reconocimiento se genera antes de la llegada del mensaje de datos a su destino final, se debe modificar el encaminador DRB para detectar que la latencia que ha sufrido un mensaje supera el valor umbral y generar el mensaje de reconocimiento. La Figura 4-35 muestra las modificaciones necesarias en el encaminador DRB para soportar esta función. Se debe incluir un módulo GMR de detección de la latencia y de generación del mensaje de reconocimiento en caso necesario. Este módulo contiene un registro que almacena el valor umbral y su función es monitorizar cada mensaje que se inyecta en un canal de salida, comparar el campo de latencia con el valor umbral y en el caso de que supere el umbral definido, extrae del mensaje de datos el mensaje de reconocimiento y lo inyecta en la cola de inyección de mensajes.

Las otras funciones de DRB, configuración de los metacaminos y selección de caminos multipaso, como ya se ha comentado, se implementan en la interfaz de red de cada nodo que inyecta mensajes. En este caso, se debe implementar, por "*hardware*", "*software*", o en una forma mixta de "*firmware*", los algoritmos presentados para cada caso.

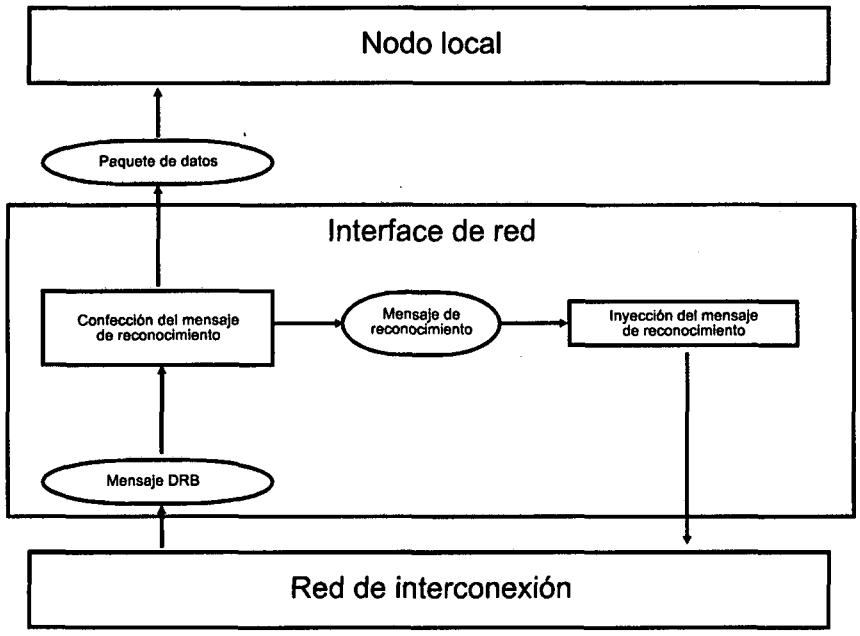


Figura 4-34 Interface de red DRB en los nodos destino

La Figura 4-36 muestra un esquema de bloques que incluye los elementos descritos: El algoritmo de selección de caminos multipaso elige un camino multipaso de entre los que componen el metacamino y, cuando el nodo inyecta un nuevo mensaje, concatena los destinos intermedios a la cabecera del paquete para formar el paquete DRB, el cual será normalmente inyectado en la red de interconexión. El otro módulo, configuración de los metacaminos, recoge los mensajes de reconocimiento que llegan al nodo con la información de latencia que sufren los caminos multipaso y configura los metacaminos para balancear las comunicaciones según el algoritmo de configuración de metacaminos presentado.

Tal y como se han presentado esos algoritmos, se observa que implican la ejecución secuencial de una serie fija de pasos simples sin iteraciones. Por esta razón sus requerimientos computacionales son mínimos. La configuración de los metacaminos se ejecuta cada vez que se recibe una latencia en el nodo origen, pero no afecta a la inyección de nuevos mensajes. La selección de un camino multipaso puede realizarse en avance para el próximo mensaje que se inyecte, por lo que su ejecución tampoco retarda la inyección de nuevos mensajes. Esta inyección supone, entre otras tareas, la composición del cabecera multidestino mediante la concatenación de los diferentes destinos intermedios.

Respecto a la memoria que requieren los algoritmos de configuración de metacaminos y de selección de caminos multipaso, se observa en los algoritmos presentados que se necesita almacenar, para cada par fuente destino, la latencia del metacamino y los nodos intermedios y la latencia de cada uno de los caminos multipaso

que lo componen. Esta es una información de tamaño conocido y configurable por parte del diseñador.

El método DRB ha sido diseñado intentando mantener el incremento de costes al mínimo y sobre todo teniendo en mente no incrementar el camino crítico de los encaminadores. Algunas de las funciones de DRB, como se ha visto, no se implementan en los encaminadores, sino en las interfaces de la red.

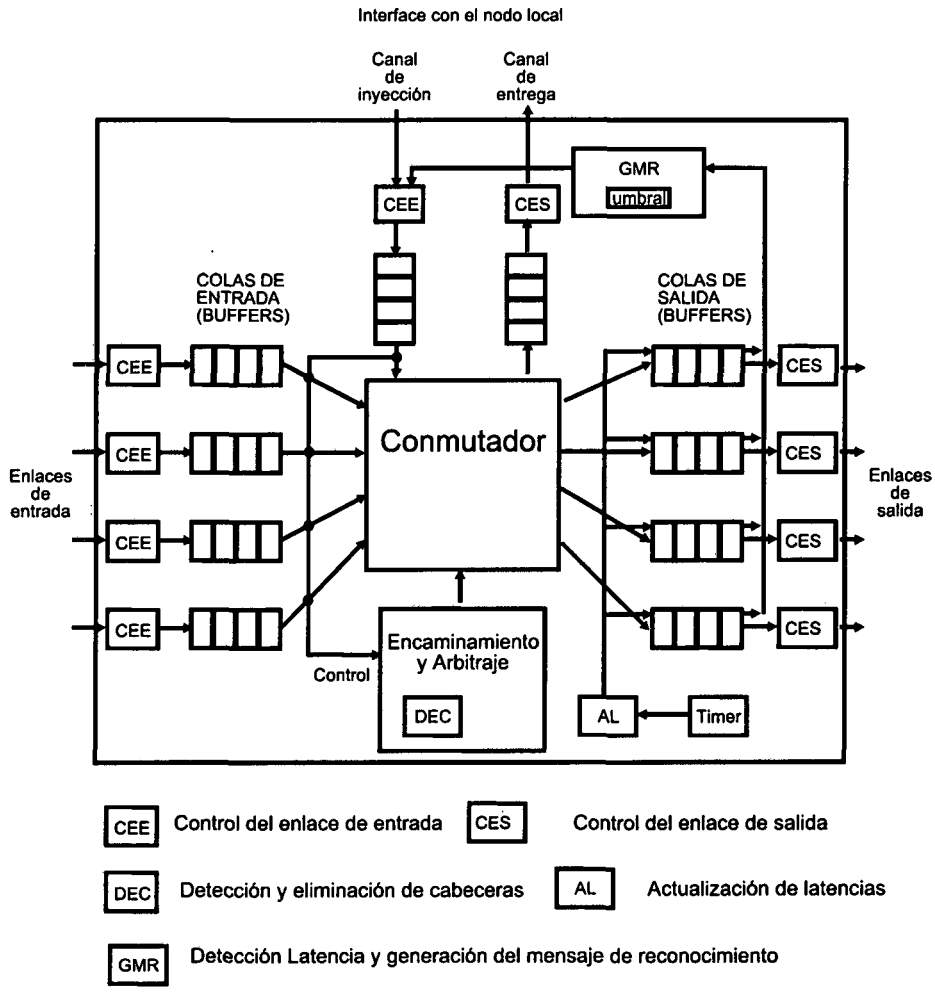


Figura 4-35 Estructura del encaminador DRB con informe temprano de la latencia

Además, en DRB no existe un intercambio periódico de información, por lo que en ausencia de tráfico, no existe ningún "*overhead*" añadido. Asimismo, presenta la característica que, bajo cargas de tráfico pequeñas o moderadas, la actividad de monitorización es mínima y los mensajes recorren los caminos mínimos determinados por el encaminamiento estático. La función de registro de la latencia se produce cuando un paquete se encola en un encaminador y va a permanecer bloqueado un cierto tiempo, por lo tanto no supone un retraso en el avance de los paquetes en tránsito. La función de

implementación de los caminos multipaso supone dos decisiones de encaminamiento extra cuando el paquete llega a cada uno de los nodos destino intermedios. Cuando el paquete llega al nodo destino, se debe examinar el bit que informa si hay mas cabeceras y, en caso afirmativo, volver a encaminar con el siguiente destino. En caso negativo, se encola el paquete en la cola de entrega de paquetes del nodo local. Esta acción se produce sólo dos veces a lo largo de todo el camino y es independiente de la longitud del mismo por lo que es una cantidad constante y pequeña respecto del número de decisiones de encaminamiento que se realizan para transportar el mensaje.

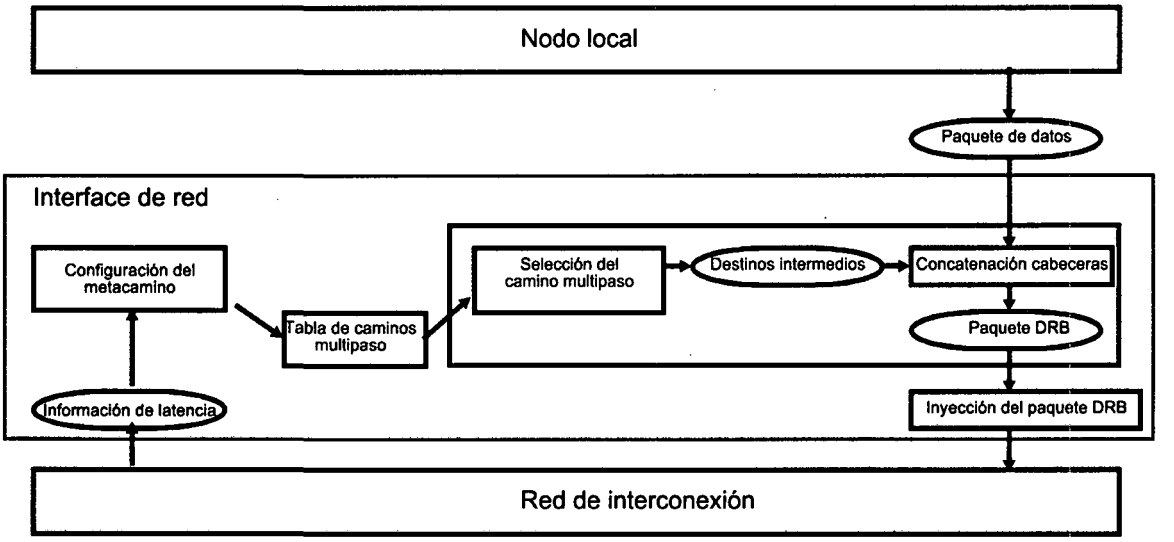


Figura 4-36 Interface de red DRB en los nodos fuente

Por otro lado, DRB es un sistema totalmente distribuido donde no se depende de un servidor central o de una jerarquía de encaminadores, sino que cada par fuente-destino es capaz de operar independientemente de los demás. Como puede observarse del análisis realizado, las actividades de monitorización, configuración y selección de caminos multipaso se ejecutan un número de veces que es proporcional al número de canales de la aplicación y al número de mensajes enviados. Por esta razón, cuando hay poca carga, el sistema no introduce casi "*overhead*", y cuando la carga de comunicaciones se incrementa, y necesita ser balanceada, es cuando actúa DRB.

4.8 Conclusiones

Este capítulo ha presentado el método DRB desarrollado como aportación principal de este trabajo de tesis. Como se ha descrito, DRB es un método de balanceo de las comunicaciones en redes de interconexión que tiene como objetivo principal mantener la latencia próxima a su valor intrínseco (propio de la red) y aumentar el porcentaje de uso de la red. Además, en el capítulo se ha analizado el comportamiento de la redes de interconexión y se han deducido las causas de la problemática que surge

con la latencia de los mensajes. Se ha visto la influencia de aspectos como el número de mensajes que colisionan, su frecuencia o su longitud. También se ha analizado las implicaciones de esta problemática en las aplicaciones paralelas, tanto de cómputo intensivo, como con características multimedia. A partir de este análisis, se han establecido los objetivos de diseño de una red de interconexión de propósito general para un computador paralelo y se ha señalado al balanceo de las comunicaciones como el método para conseguir los objetivos de diseño de las redes de interconexión. La parte fundamental del capítulo se ha dedicado a la explicación con profundidad de la propuesta DRB. Se han descrito sus dos componentes, la definición de los caminos alternativos y la configuración dinámica de los mismos, y se ha mostrado la implementación de este mecanismo en el encaminador DRB.

En el siguiente capítulo 5, se presenta el dimensionamiento de los metacaminos y se hace un estudio de sus características en función del la topología y el número de nodos de la red de interconexión.

Capítulo 5 Evaluación de DRB: Características de los metacaminos

5.1 Introducción

En este capítulo se presenta la caracterización de una serie de propiedades topológicas de las redes de interconexión, ligadas con los caminos para los mensajes, para su uso con DRB: las características de los metacaminos. Se explican las topologías y el encaminamiento utilizado para hacer las diferentes mediciones, así como una explicación de la forma cómo se realizaron los experimentos. También se exponen los resultados más interesantes de toda esta investigación y finalmente se hace un análisis de dichos resultados. Una versión completa de esta experimentación fue presentada en [57] correspondiente al trabajo de investigación de I. Garcés Botacio realizado en el marco de este trabajo de tesis.

El objetivo principal de las pruebas es evaluar por medio de la simulación los parámetros de los metacaminos de DRB, sin presencia de tráfico en la red. Con esta

experimentación nos interesa medir tres aspectos. El primero es la latencia en ausencia de contención de los mensajes cuando se usa el método DRB, que en el supuesto de un único mensaje en la red depende de la longitud del camino, es decir, pretende medir el alargamiento extra provocado en los caminos que recorren los mensajes cuando se utilizan una cierta configuración de supernodos y caminos multipaso. Este alargamiento es la longitud del metacamino definido por la Ecuación 4-9. Segundo, el ancho de banda extra que pone a disposición el método DRB para que se envíen los mensajes entre un nodo fuente y un nodo destino. Este segundo aspecto es, por decirlo con otras palabras, el número de caminos alternativos o anchura del metacamino, determinados por la Ecuación 4-8, según la definición hecha en el capítulo anterior. Tercero, el aprovechamiento que se hace de los enlaces de los nodos (grado), es decir, medir en que medida se utilizan los enlaces disponibles de cada nodo.

Para realizar las medidas, se ha utilizado el simulador funcional de redes de interconexión presentado en el capítulo 2. Con esta herramienta, que simula las topologías de las redes de interconexión en estudio, se han medido los parámetros antes mencionados.

El resto del capítulo esta organizado de la siguiente manera. La sección 2 describe el diseño de los experimentos realizados, incluyendo la estructura de los supernodos configurados, el diseño de las pruebas y los parámetros medidos. A continuación, la sección 3 presenta y analiza los resultados obtenidos. En esta sección, primero se muestran detalladamente los valores numéricos de los resultados para un tamaño de topología y a continuación se grafican los resultados para diversos tamaños de las topologías elegidas. Esta sección concluye con el estudio de la escalabilidad del método DRB. Finalmente, la sección 4 presenta las conclusiones del capítulo.

5.2 Diseño de los Experimentos

Como hemos visto en capítulos anteriores existe una amplia gama de topologías. Las topologías seleccionadas para hacer la experimentación y evaluación de DRB fueron los *n-cubos k-arios* (toros e hipercubos), elegidos por ser los más utilizados en computadores paralelos, y las redes *"midimew"*, por presentar unas características topológicas interesantes, como es el hecho de tener una distancia promedio menor que los toros para el mismo número de nodos.

Una vez determinadas las topologías para evaluar el método DRB, es necesario plantearse cuáles son los valores de los parámetros a medir, y la forma de lograrlos mediante el diseño de pruebas específicas. Esta sección se dividirá en tres partes:

primero, se presentará las estructuras de supernodos configurados para el método DRB; segundo, el diseño de las pruebas realizado; y tercero, las medias y cálculos extraídos.

5.2.1 Estructura de los supernodos

Lo primero que determinaremos es la forma específica como se construyen los supernodos para las topologías elegidas en la experimentación realizada. Los métodos fueron aplicados a las tres redes de interconexión mencionadas variando la topología, la configuración y el tamaño.

En la Tabla 5-1 se muestran las estructuras elegidas para realizar los supernodos de acuerdo con el método DRB. Las filas de la tabla representan cada uno de los métodos evaluados. En este cuadro se añaden como método de comparación, el encaminamiento estático y el encaminamiento aleatorio.

El encaminamiento estático tiene supernodos vacíos tanto para el nodo fuente como el destino (no utiliza nodos intermedios), y el encaminamiento aleatorio tiene como supernodo fuente todos los nodos de la red. Ubicados entre estos dos casos extremos, tenemos los casos formados por el método DRB, según se elija los supernodos de subtopologías o de áreas de gravedad. En la última columna se muestra la formación de los supernodos. Como se puede observar, para cada topología fueron probados todos los métodos con la construcción específica del supernodo.

En esta experimentación, se han considerado los métodos de construcción de los metacaminos de subtopologías y áreas de gravedad. Para los primeros, los métodos de subtopologías, se ha configurado un determinado supernodo fuente según la definición y un supernodo destino mínimo canónico, es decir, un solo nodo. Para los métodos de área de gravedad se ha considerando tanto configurar el supernodo destino como canónico (llamado área de gravedad simple porque sólo el nodo fuente configura un área de gravedad) o como no canónico (llamado área de gravedad doble).

En el método de subtopologías, el supernodo depende de la topología de la red de interconexión. Para un toro 2D o una *"midimew"* el supernodo es una fila o una columna de la que se varía su tamaño. Para un hipercubo, es un hipercubo de dimensión menor, que se va reduciendo sucesivamente. En un toro 3D la reducción se puede hacer por filas, columnas o por planos. La reducción del tamaño de una fila de tamaño m o de una columna de tamaño n, para un toro o una *"midimew"* se hace dividiendo por el incremento sucesivo de potencias de dos hasta llegar a m o n, respectivamente, como se puede ver en la Tabla 5-1. En el caso de un hipercubo, se subdivide la dimensión de igual manera, hasta llegar al límite de la mitad de la dimensión.

Método de Encaminamiento		Topología	Estructura de los Supernodos	
			Fuente/Destino	Tamaño y Estructura
Encaminamiento Estático		Toro2D, *Midimew*, Toro 3D, Hipercubo	Supernodo Fuente : Nodo Fuente	1
			Supernodo Destino: Nodo Destino	1
DRB	Supernodos De Sub-topologías	Toro 2D, Toro3D *Midimew* nxm	Supernodo Fuente: Fila, Columna, Plano o Hipercubo	m/i; $i = (2^0, 2^1, 2^2, ..., {}^m/_2)$
				n/i; $i = (2^0, 2^1, 2^2, ..., {}^n/_2)$
		Hipercubo 2^{Dim}		$2^{Dim/2}/i$; $i = (2^0, 2^1, ..., 2^{Dim/2})$
			Supernodo Destino: Nodo Destino	1
	Área de Gravedad Simple	Toro2D Toro3D *Midimew* Hipercubo	Supernodo Fuente: Area de Gravedad	$d_G = (1, 2, ..., {}^{Max.\ Dist.}/_2)$
			Supernodo Destino: Nodo Destino	1
	Área de Gravedad Doble		Supernodo Fuente: Area de Gravedad	$d_G = (1, 2, ..., {}^{Max.\ Dist.}/_2)$
			Supernodo Destino: Area de Gravedad	$d_G = (1, 2, ..., {}^{Max.\ Dist.}/_2)$
Encaminamiento Aleatorio		Toro 2D, Toro 3D *Midimew*, Hipercubo	Supernodo Fuente: Area de Gravedad	nxm ó 2^{Dim}
			Supernodo Destino: Nodo Destino	1

Tabla 5-1. Experimentación para cada método y topología

En los métodos de áreas de gravedad, el tamaño del área se tomó desde $d_G = 1$ hasta la mitad de la distancia máxima de la red para el caso de un par fuente-destino a distancia máxima, que es el tamaño de d_G donde las áreas no se superponen. En el caso de pares fuente-destino cuya distancia sea menor de la distancia máxima de la red, las áreas de gravedad se limitan en su tamaño de manera que nunca se superpongan el área del nodo fuente con la del destino. Obsérvese que sólo en el método de dobles áreas de gravedad hay dos supernodos de nodos intermedios.

5.2.2 Diseño de las pruebas

Una vez determinada la forma de construir los supernodos, se necesita el diseño de las pruebas. Como se ha mencionado en la introducción, se ha utilizado el simulador funcional desarrollado en este trabajo de tesis. La experimentación se diseñó de la siguiente manera: consistió en simular series de envíos consecutivos de mensajes tomando como nodo fuente un nodo fijo y como nodo destino todos los otros nodos de la red. Nos interesan transferencias de mensajes que impliquen el uso de la red, por lo tanto, se excluye el caso de nodos que se envíen mensajes a sí mismos. Antes de enviar un nuevo mensaje, se espera a que el anterior haya llegado a su destino, para evitar las colisiones entre mensajes, ya que se quiere medir solamente distancias recorridas.

Para cada par fuente-destino [(0, 1)...(0, N-1)] existe un conjunto de nodos intermedios contenidos en uno o dos supernodos, centrados en el nodo fuente (0...L-1) o centrados en el nodo destino (0...M-1) seleccionados de acuerdo al método deseado, como se muestra en la Figura 5-1(a). Cabe recordar, que debido a la propiedad de simetría de las topologías, los resultados para un nodo son los mismos para cualquier nodo de la red tomado como nodo fuente. Es por esta razón que la experimentación se ha realizado para un único nodo fuente contra todos los nodos destino.

También, como nuestro objetivo es analizar la distribución de los caminos, en el conjunto de nodos destino se excluyen los que forman parte del supernodo, porque no tiene sentido ir a nodos intermedios localizados en la misma zona que el nodo destino.

5.2.3 Parámetros medidos y calculados

Este apartado especifica cómo se van a medir los parámetros involucrados en la evaluación. Las siguientes medidas fueron tomadas para poder evaluar los parámetros de los supernodos (alargamiento del camino y ancho de banda usado) tal y como se muestra en la Figura 5-1 (b):

- ✓ La distancia promedio de todos los caminos definidos por el supernodo para cada par fuente-destino, es decir, la longitud del metacamino formado por los supernodos.

- ✓ Los enlaces (entrada/salida) utilizados por los nodos y la cantidad de nodos intermedios proporcionados por el supernodo elegido, para encontrar la ganancia lograda con respecto al ancho de banda. El parámetro del número de nodos es una medida indirecta del ancho de banda que DRB proporciona para comunicar un par de nodos fuente-destino porque es el número de nodos que potencialmente pueden

estar dedicados a comunicar paquetes de ese par fuente-destino, es decir, en un momento dado, todos el ancho de banda de cada uno de esos nodos podría estar transmitiendo información perteneciente a ese par fuente-destino.

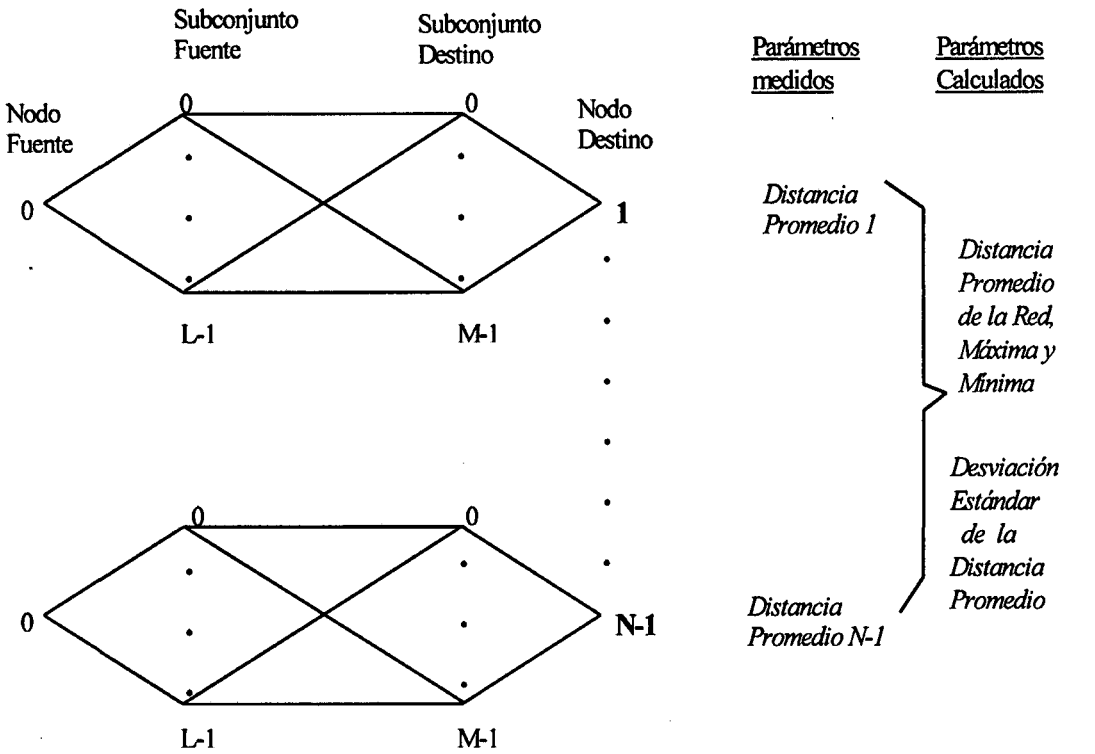


Figura 5-1 Experimentación para la evaluación de los metacaminos

(Supernodo Fuente: L nodos diferentes de la red.

Supernodo Destino: M nodos diferentes de la red. N = Número total de nodos en la red)

A partir de los valores anteriores, se han calculado las distancias promedio, mínima, máxima y la desviación estándar de las anteriores distancias para toda la topología (Figura 5-1 (c)). Esta distancia promedio se puede considerar la distancia media de la topología con el encaminamiento DRB, que es similar a la distancia media física con encaminamiento estático mínimo. Esta distancia promedio determina el valor mínimo de la latencia en ausencia de contención.

Estos valores para cada método de encaminamiento DRB se comparan porcentualmente con los del encaminamiento estático y se extrae el porcentaje de

alargamiento del camino y el ancho de banda utilizado medido como el número de nodos pertenecientes al supernodo.

Los métodos fueron aplicados a redes toros 2D y 3D, *"midimew"* e hipercubos, con rangos desde 8 hasta un máximo de 64K nodos para hipercubos y toros 3D, y desde 9 hasta 64K nodos para Toros 2D y *"midimews"*.

5.3 Resultados y Análisis

En este punto se presentan los resultados extraídos de toda la experimentación realizada que se ha descrito en el punto anterior. Debido al gran volumen de datos obtenidos y para facilitar su comprensión y análisis, hemos dividido esta sección en tres apartados. El primero presenta los resultados exhaustivos para todas las topologías seleccionadas de un único tamaño de 1024 nodos. Estos resultados se presentan en forma de tabla para observar los valores numéricos con precisión. El siguiente apartado presenta los resultados obtenidos para todos los tamaños de topología evaluados. Esta vez los datos se presentan de forma gráfica, lo cual facilita la comparación visual y el análisis de tendencias. El último apartado se fija en la escalabilidad del método DRB, es decir, que el alargamiento de los caminos y el ancho de banda proporcionado por los metacaminos crezca de manera proporcional al tamaño de la red. Para ello, presenta unas gráficas sobre el tamaño de la red.

5.3.1 Resultados para 1024 nodos

Este apartado muestra los resultados obtenidos para un ejemplo de cada red, todas ellas el mismo tamaño (1024 nodos). Los resultados obtenidos durante la experimentación se muestran en tres tablas, donde se observa los diferentes tamaños de supernodos que fueron evaluados para los métodos de Subtopologías y de Áreas de Gravedad, subdivididos en Áreas de Gravedad Simple (SGA) y Áreas de Gravedad Dobles (DGA). La Tabla 5-2 presenta los resultados para un toro bidimensional regular de 1024-nodos (32x32). La Tabla 5-3 para una *"midimew"* de 1024-nodos con una configuración de 34 x 34. La Tabla 5-4 muestra los resultados para un hipercubo de dimensión 10, también de 1024 nodos. En [57] se muestran las tablas para todas las topologías y tamaños sobre los que se hizo la experimentación.

Cada tabla contiene, para cada método de supernodo las siguientes medidas. Una interpretación cualitativa de estas medidas nos da una idea de las propiedades a medir.

- ✓ La distancia promedio de la red, tal y como se ha definido anteriormente. La distancia promedio es una medida de la latencia sin tráfico en la red.

- ✓ El alargamiento porcentual con respecto a la distancia del encaminamiento estático. El porcentaje de alargamiento es útil para comparar esta latencia con la distancia mínima proporcionada por el encaminamiento estático.

- ✓ La desviación estándar de las distancias. La desviación estándar muestra la uniformidad del método con respecto al alargamiento del camino

- ✓ El número de nodos intermedios utilizados por el supernodo. El número de nodos intermedios muestra el ancho de banda extra obtenido por la creación de caminos alternativos

- ✓ Una clasificación graduada dependiendo de la utilización de los enlaces. Si el método de encaminamiento sólo utiliza un enlace de cada nodo, se clasifica como un uso del grado "mínimo", si utiliza solo los enlaces que pertenecen a una de las dimensiones, se califica como "bajo"; si se utilizan todos los enlaces del nodo, se califica como "alto"; si se utiliza un valor entre ambos("bajo" y "alto"), se califica como "medio". La utilización de los enlaces muestra si se logra un uso total del grado del nodo.

También se incluyen los resultados de los métodos estático y aleatorio para cada topología como información comparativa.

5.3.1.1 Análisis de la Tabla 5-2, la Tabla 5-3 y la Tabla 5-4

A continuación presentamos algunas observaciones sobre el estudio de los métodos de supernodos de subtopologías y de áreas de gravedad extraídas de las tablas anteriores.

En todos los casos, la distancia promedio (segunda columna de la Tabla 5-2, la Tabla 5-3 y la Tabla 5-4) está directamente relacionada con el número de nodos del supernodo (quinta columna). Por lo tanto, usar DRB incrementa la distancia recorrida. Pero, por otro lado, el número de nodos del supernodo (ancho de banda extra), como se esperaba, crece proporcionalmente al tamaño del supernodo, lo que implica que el ancho de banda extra obtenido siempre crece con el tamaño del supernodo. Como se puede ver, todos los métodos guardan relación en los rangos de las distancias promedio respecto el tamaño del supernodo (primera y segunda columnas), pero con gran diferenciación en el ancho de banda utilizado (quinta columna), que en el caso de los métodos de áreas de gravedad, el número de nodos es mucho mayor.

TORO (2D) 1024 nodos (32 X 32)

MÉTODO	Distancia Promedio (Latencia)	Alargamiento del camino (%)	Desviación Estándar de las Distancias	Nº de Nodos intermedios del Supernodo	Utilización de los Enlaces
Aleatorio	32.0	100%	0	1024	Alto
Supernodos de Subtopología (Dimensión-Tamaño)					
Fila32	24.3	51.47%	4.48	32	Bajo
Fila16	23.8	48.34%	5.35	16	Bajo
Fila8	19.8	23.37%	6.11	8	Bajo
Fila4	17.8	10.88%	6.36	4	Bajo
Fila2	16.8	4.63%	6.44	2	Bajo
Área de Gravedad Simple ($SGAd_G$)					
SGA16	29.1	81.71%	1.40	543	Medio
SGA8	22.9	43.37%	4.22	145	Medio
SGA4	19.4	21.26%	5.72	41	Medio
SGA2	17.7	10.61%	6.25	13	Medio
SGA1	16.9	5.35%	6.43	5	Medio
Doble Área de Gravedad ($DGAd_G$)					
DGA8/8	30.3	89.38%	2.42	290	Alto
DGA4/4	23.0	43.94%	4.87	82	Alto
DGA2/2	19.4	21.25%	6.00	26	Alto
DGA1/1	17.7	10.63%	6.38	10	Alto
Estático	16.0	0%	6.54	0	Mínimo

Tabla 5-2 Resultados de experimentación Estática para Toro 2D

MIDIMEW 1024 nodos (34 X 34)

MÉTODO	Distancia Promedio (Latencia)	Alargamiento del camino (%)	Desviación Estándar de las Distancias	Nº de Nodos intermedios del Supernodo	Utilización de los Enlaces
Aleatorio	30.1	100%	0	1024	Alto
Supernodos de Subtopología (Dimensión-Tamaño)					
Fila34	23.7	57.28%	2.29	34	Bajo
Fila17	19.5	29.03%	3.86	17	Bajo
Fila8	17.2	13.81%	4.86	8	Bajo
Fila4	16.1	6.89%	5.18	4	Bajo
Fila2	15.6	3.41%	5.29	2	Bajo
Área de Gravedad Simple ($SGAd_G$)					
SGA16	27.1	79.69%	0.60	545	Medio
SGA11	24.2	60.59%	1.45	265	Medio
SGA8	21.9	45.03%	2.71	145	Medio
SGA5	19.3	27.99%	4.01	61	Medio
SGA4	18.4	22.32%	4.38	41	Medio
SGA2	16.7	11.19%	4.99	13	Medio
SGA1	15.9	5.65%	5.20	5	Medio
Doble Área de Gravedad ($DGAd_G$)					
DGA5/5	23.8	57.66%	2.65	122	Alto
DGA2/2	18.4	22.26%	4.71	26	Alto
DGA1/1	16.7	10.94%	5.14	10	Alto
Estático	15.1	0%	5.33	0	Mínimo

Tabla 5-3 Resultados de experimentación Estática para Midimew

HIPERCUBO 10 D (2^{10})

MÉTODO	Distancia Promedio (Latencia)	Alargamiento del camino (%)	Desviación Estándar de las Distancias	Nº de Nodos intermedios del Supernodo	Utilización de los Enlaces
Aleatorio	10.0	100%	0	1024	Alto
Supernodos de Subtopología (Dimensión)					
5D	7.6	51.49%	1.04	32	Bajo
4D	7.0	40.86%	1.17	16	Bajo
3D	6.5	30,47%	1.29	8	Bajo
2D	6.0	20,25%	1.39	4	Bajo
1D	5.5	10,14%	1.48	2	Bajo
Área de Gravedad Simple ($SGAd_G$)					
SGA5	9.3	86.51%	0.16	638	Medio
SGA4	8.7	73.63%	0.33	386	Medio
SGA3	7.8	56.85%	0.59	176	Medio
SGA2	6.9	37.97%	0.91	56	Medio
SGA1	5.9	18.79%	1.25	11	Medio
Doble Área de Gravedad ($DGAd_G$)					
DGA2/2	8.8	75.46%	0.53	112	Alto
DGA1/1	7.7	54.44%	1.27	22	Alto
Estático	5.0	0%	1.57	0	Mínimo

Tabla 5-4 Resultados de experimentación Estática para Hipercubo 10D

Se observa que la topología de hipercubo es capaz de conseguir un mayor uso del número de nodos (hasta 638 nodos para el método SGA5 con un alargamiento del 86.51%) que las topologías de toro (543 nodos para el método SGA16, que supone un alargamiento del 81.71%) y "*midimew*" (545 nodos para el método SGA16 con un alargamiento del 79.69%). Las topologías toro y "midimew" son muy similares entre ellas, presentando un comportamiento ligeramente superior la "*midimew*". Aparte del amplio rango de alternativas ofrecidas por los métodos, que se encuentran entre los métodos estático y aleatorio, se puede observar también que los métodos de áreas de gravedad ofrecen una razón entre el ancho de banda extra y la latencia más alta que los métodos de subtopologías, para cada una de las gráficas.

Se puede ver que con determinados tamaños de supernodos (para toro Fila16, Fila8, Fila4, Fila2, SGA8, SGA4, SGA2, SGA1, DGA4/4, DGA2/2, DGA1/1) la penalización sobre el alargamiento del camino (menor del 50%) con respecto al método estático es insignificante y, en cambio, el ancho de banda utilizado para esos métodos (número de nodos del supernodo) varía desde 2 hasta 638, lo cual son valores muy altos. Además, se puede lograr aumentar el uso de los enlaces de una utilización mínima a valores medio o altos, y dependiendo de los requerimientos de latencia/ancho de banda que el usuario determine como necesarios.

En las tablas anteriores, para topologías toro y *"midimew"* (Tabla 5-2 y Tabla 5-3), si se miran las segunda y tercera columnas (distancia promedio y el alargamiento porcentual del camino con respecto al método estático) respecto la quinta (nodos del supernodo), se demuestra que el crecimiento del alargamiento es proporcional con relación al tamaño del supernodo para todos los métodos. Los supernodos mayor y menor de cada método (Fila32, Fila2, SGA16, SGA1, DGA8/8, DGA1/1 para toro y Fila34, Fila2, SGA16, SGA1, DGA5/5, DGA1/1 para "*midimew*") son excepciones que muestran un alargamiento menor, y por lo tanto, son mejores.

Contrariamente a lo que sucede en toros y "*midimews*", en topologías hipercubo (Tabla 5-4), la reducción a hipercubos de tamaños menores de los supernodos, con el método de subtopologías, hace que la variación del alargamiento del camino entre un método y el siguiente (en filas consecutivas) esté determinado por un valor constante para todos los supernodos (columna "alargamiento del camino"), independientemente del tamaño del supernodo. Para los métodos de áreas de gravedad (SGA y DGA), esta variación del crecimiento del supernodo entre un tamaño de supernodo y el siguiente vuelve a ser proporcional al tamaño del supernodo.

Con relación al parámetro de uso de los enlaces por el nodo (última columna de la Tabla 5-2, la Tabla 5-3 y la Tabla 5-4), la utilización de los enlaces para el método

estático es "mínima" porque sólo utiliza un enlace predeterminado cada vez. Es "alto" para el Método Aleatorio y para el de Dobles Áreas de Gravedad porque los mensajes se distribuyen a cualquier enlace tanto del nodo fuente como del nodo destino. En el caso de Áreas de Gravedad Simples, la utilización es "media" dependiendo del tamaño del área, de la ubicación de los nodos fuente y destino y de la topología. Para los supernodos de subtopologías, es "bajo" porque normalmente en éste método se aprovecha el grado con relación al nodo fuente pero no para el nodo destino.

Con respecto a la desviación estándar, los resultados de las tablas anteriores (cuarta columna de la Tabla 5-2, la Tabla 5-3 y la Tabla 5-4) muestran que ésta crece inversamente proporcional al alargamiento. Esto significa que los métodos con bajo alargamiento son menos uniformes, de manera que a medida que crece el alargamiento, la distribución de caminos es más uniforme.

5.3.2 Resultados variando el tamaño de la red

A continuación, se muestran los resultados de todos los experimentos realizados para diversos tamaños de topologías. La relación entre la distancia promedio y el alargamiento porcentual del camino se observan mejor gráficamente, así como también la diferencia entre los supernodos formados reduciendo el tamaño de la fila/columna para topologías toro y *"midimew"*, o con las reducciones en dimensión en topologías hipercubo y las reducciones de subplanos del toro 3D.

En las Figura 5-2, Figura 5-3, Figura 5-4, Figura 1-3 y Figura 5-6 se muestran en gráficos los resultados por topologías (toro 2D, "*midimew*", toro 3D e hipercubo) para diferentes tamaños de red, mostrando la distancia promedio de los métodos de supernodos tipo subtopología con diferentes tamaños. Las Figura 5-7, Figura 5-8, Figura 5-9 y Figura 5-10 muestran los mismos resultados para los supernodos tipo área de gravedad simple. Las Figura 5-11, Figura 5-12, Figura 5-13 y Figura 5-14 muestran los resultados para los supernodos tipo área de gravedad doble.

Obsérvese, que miradas las gráficas fijando el valor del eje y (un tamaño de red) para todos los valores del eje x se observan los diferentes tamaños del supernodo para una topología, y miradas fijando el valor del eje x (un tamaño de supernodo) se muestra la escalabilidad de un método de supernodo para diferentes tamaños de la topología.

En las Figura 5-2 y Figura 5-3 se observan los resultados para toros y *"midimews"* desde 9 hasta 64K nodos y podemos ver el crecimiento proporcional de la distancia promedio a medida que aumenta el tamaño de los supernodos y la diferencia con la distancia promedio proporcionada por los métodos estático y aleatorio.

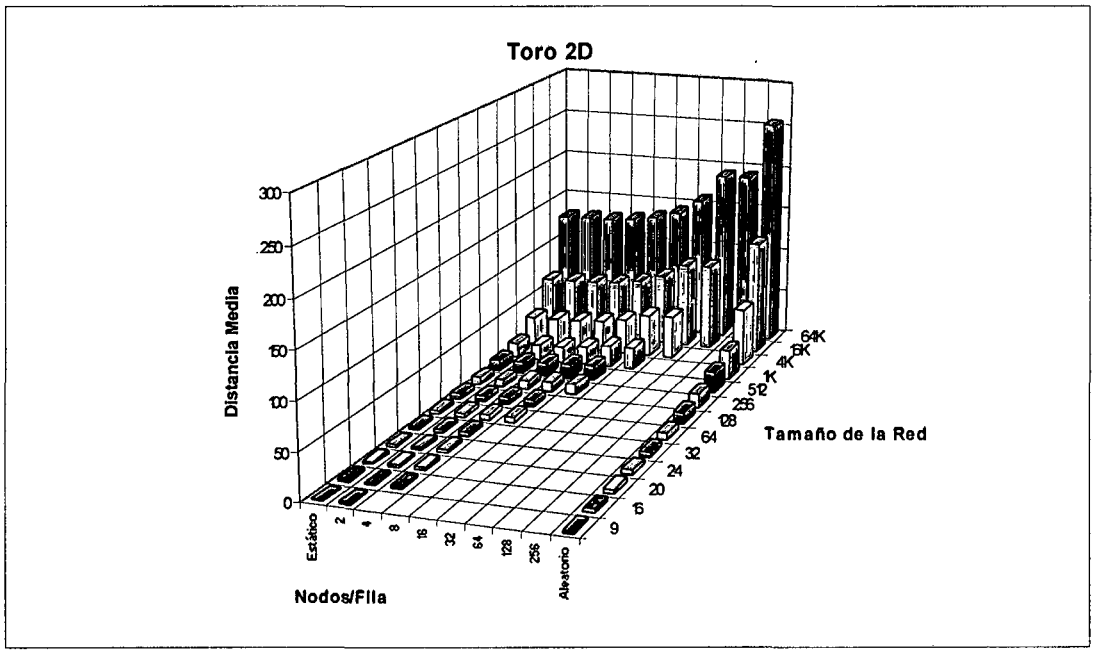


Figura 5-2 Supernodos de Subtopologías Fila n para toros 2D.

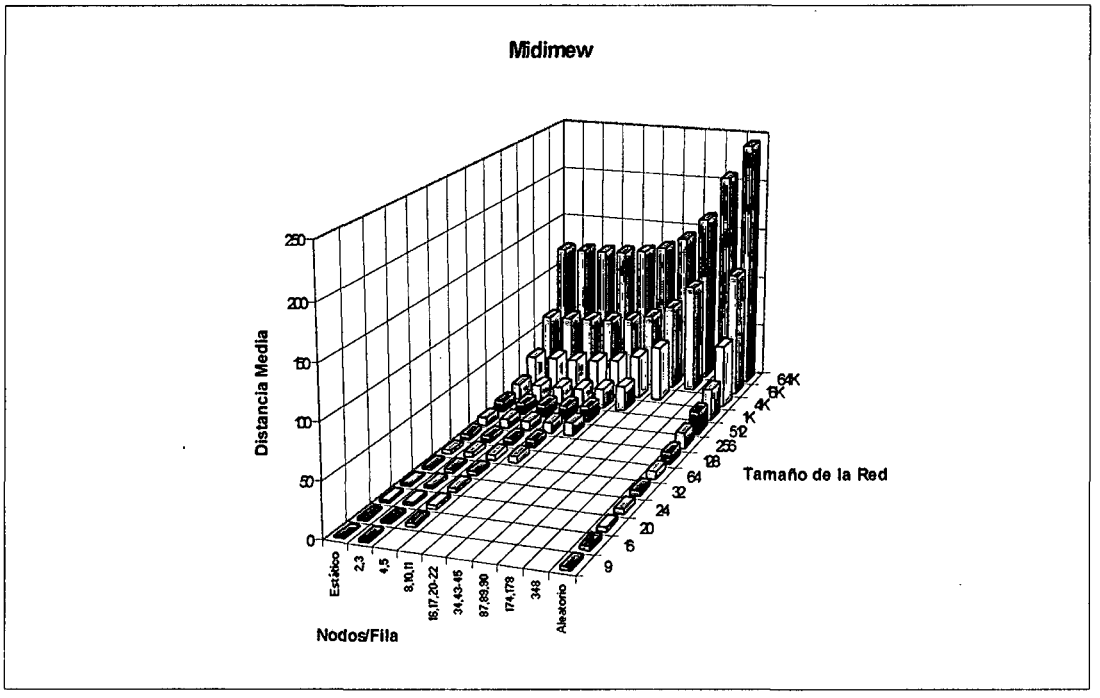


Figura 5-3 Supernodos de Subtopologías Fila n para "*Midimew*".

Para los toros 3D, el crecimiento del alargamiento es proporcional con relación al tamaño del supernodo para ambos casos de reducción en el tamaño de filas o en la dimensión (Figura 5-4 y Figura 5-5), en cambio para el hipercubo el crecimiento es constante (Figura 5-6).

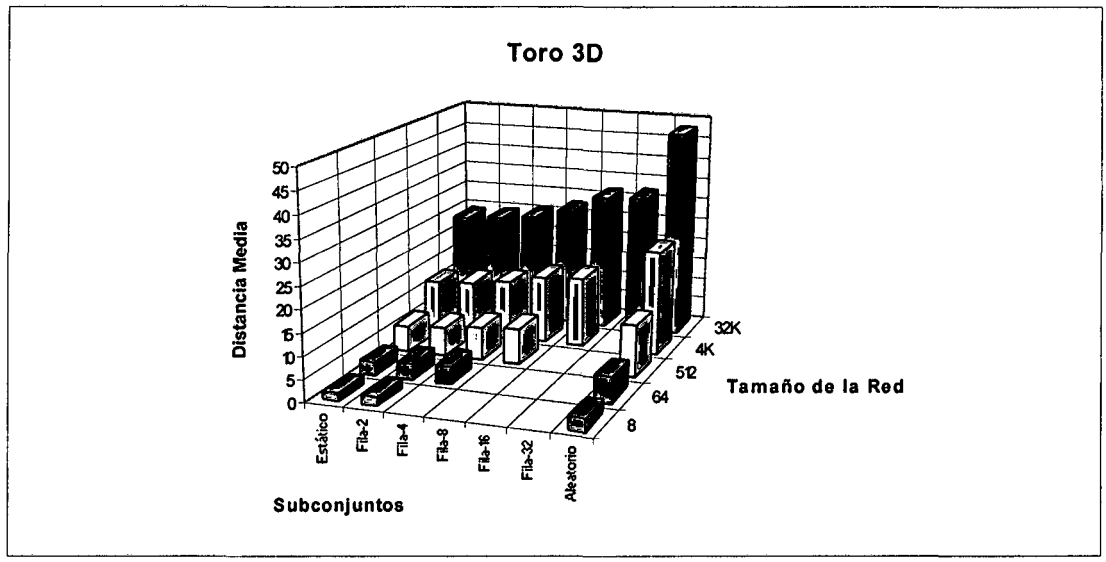


Figura 5-4 Supernodos de Subtopologías por Fila de tamaño n para toros 3D.

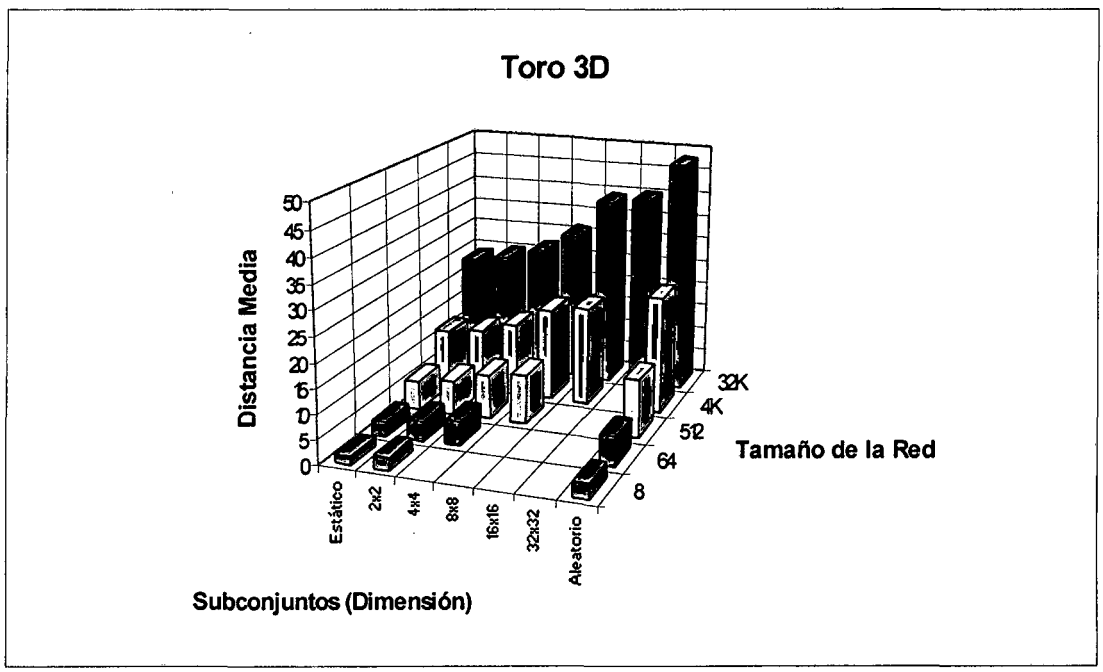


Figura 5-5 Supernodos de Subtopologías n x m para toros 3D

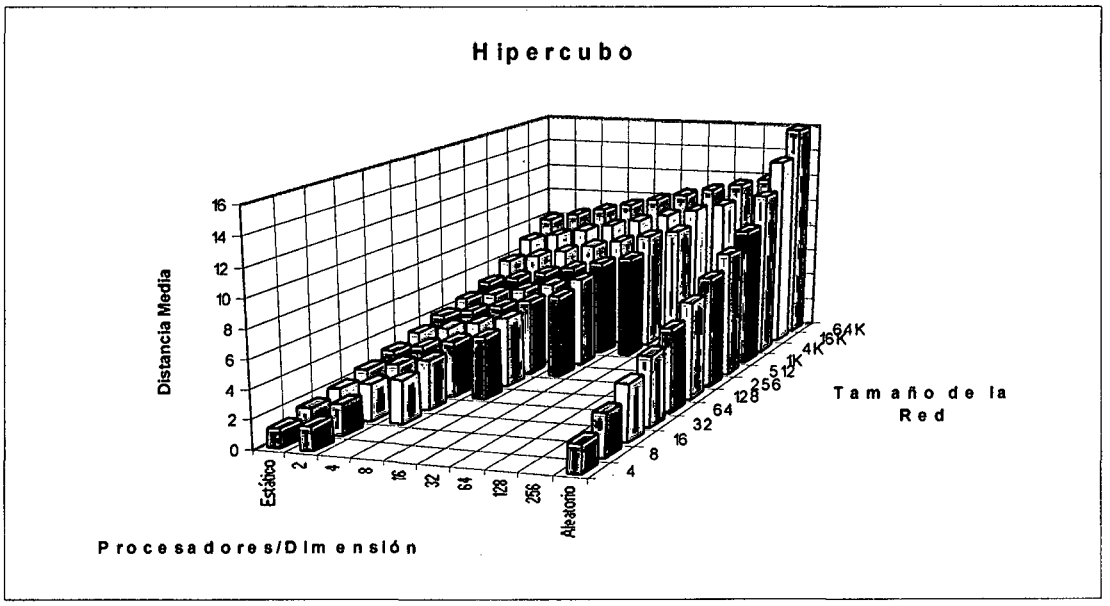


Figura 5-6 Supernodos de Subtopologías Dim/2 para Hipercubos.

Para los métodos de áreas de gravedad, se observa que el crecimiento del alargamiento es también proporcional en relación con el tamaño del supernodo. En las Figura 5-7 y Figura 5-8 se muestran los gráficos de los resultados obtenidos para las topologías toro y *"midimew"* utilizando el método de áreas de gravedad simples y se les ha añadido la comparación con el supernodo de topologías de tamaño *n*.

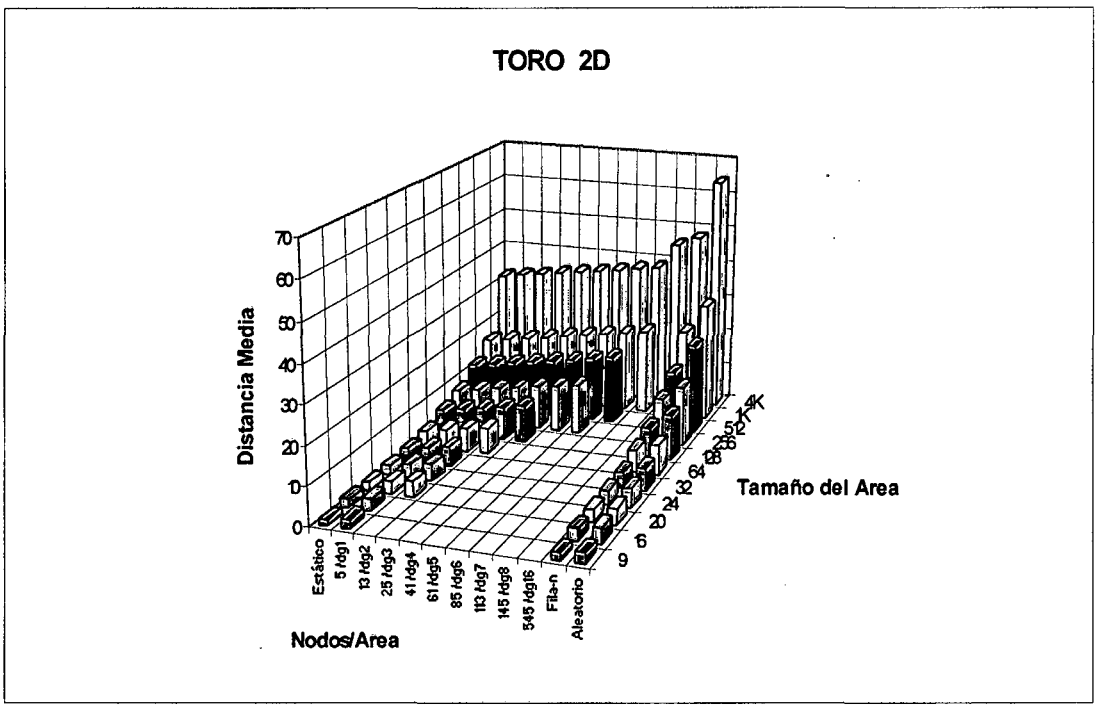


Figura 5-7 Áreas de Gravedad Simples. Toro 2D

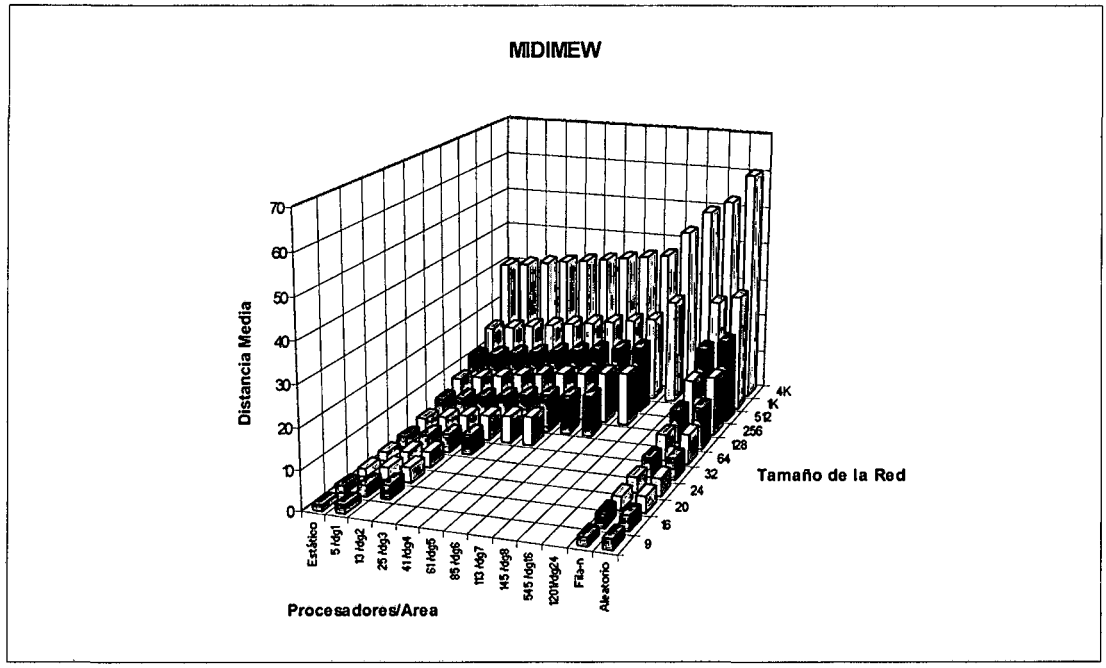


Figura 5-8 Áreas de Gravedad Simples. "*Midimew*"

En las Figura 5-9 y Figura 5-10 se observan las gráficas de los resultados obtenidos para todas las topologías hipercubo y toro 3D utilizando el método de áreas de gravedad simples, el alargamiento es dependiente del grado del nodo como se puede ver en el hipercubo, en la Figura 5-9, donde los tamaños de los supernodos son muy grandes y crecen más rápidamente que por ejemplo, para el toro 3D, en la Figura 5-10.

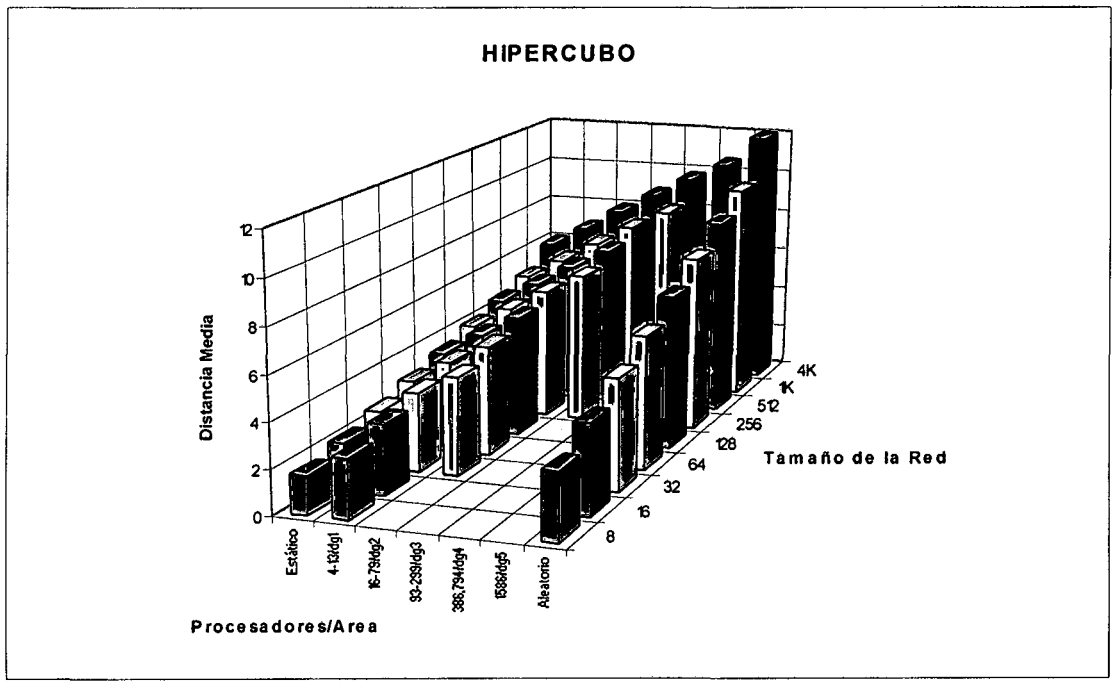


Figura 5-9 Áreas de Gravedad Simples. Hipercubo.

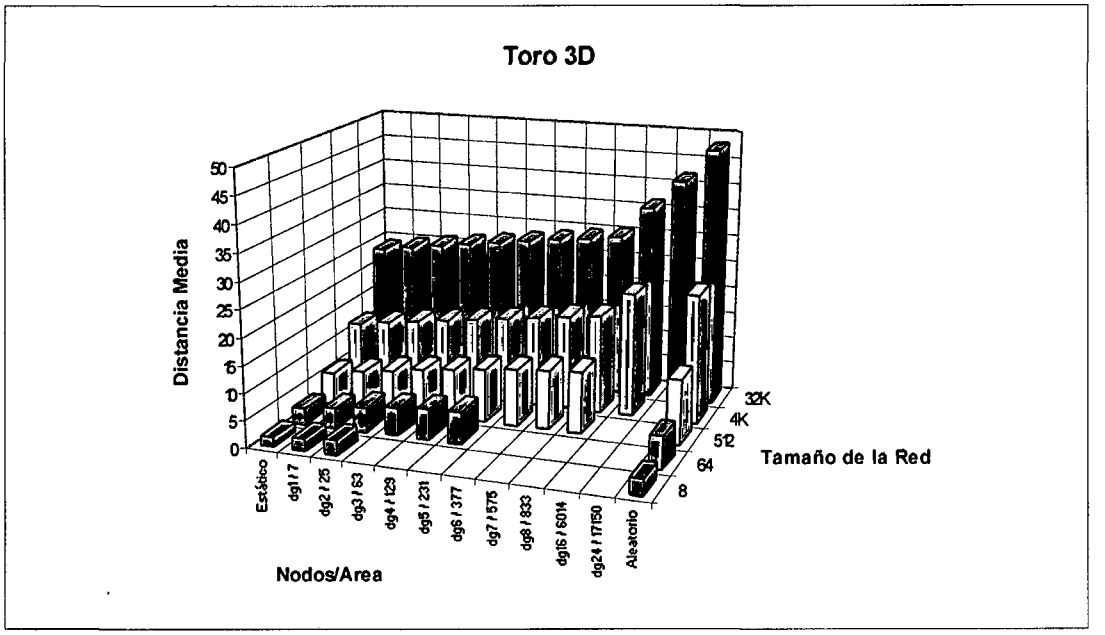


Figura 5-10 Áreas de Gravedad Simples. Toro 3D

En las Figura 5-11, Figura 5-12, Figura 5-13 y Figura 5-14 se pueden ver los gráficos de los resultados para todas las topologías utilizando el método de dobles áreas de gravedad, obteniéndose resultados similares que para las áreas de gravedad simples, pero con mayor aprovechamiento del ancho de banda.

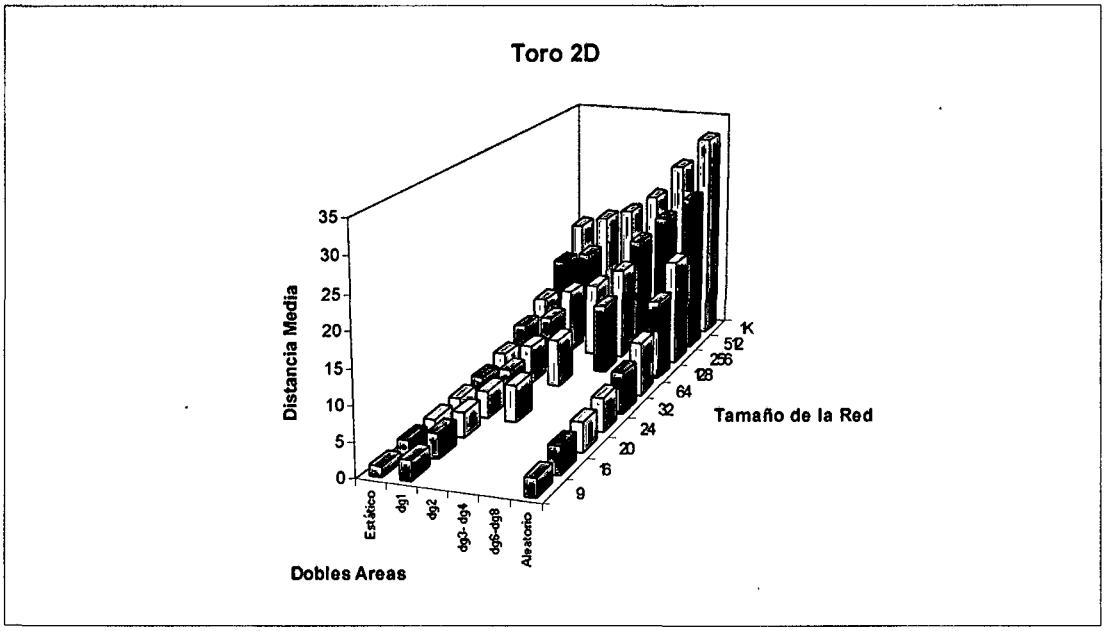


Figura 5-11 Áreas de Gravedad Dobles. Toro 2D

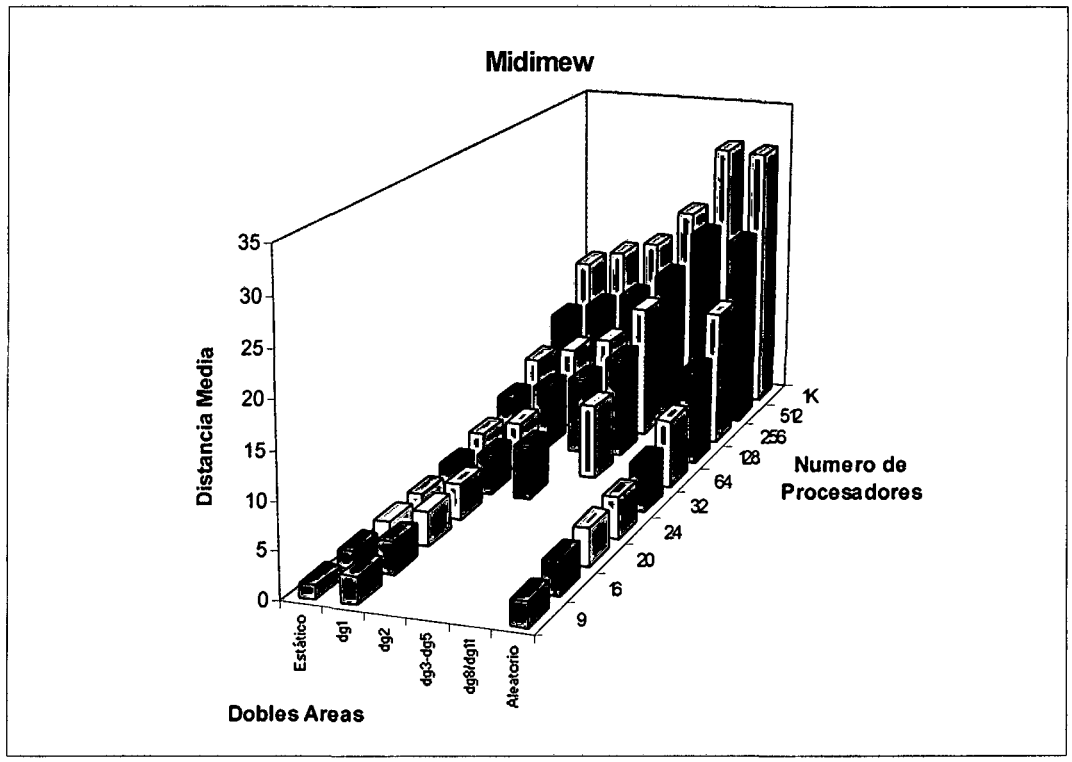


Figura 5-12 Areas de Gravedad Dobles. "*Midimew*"

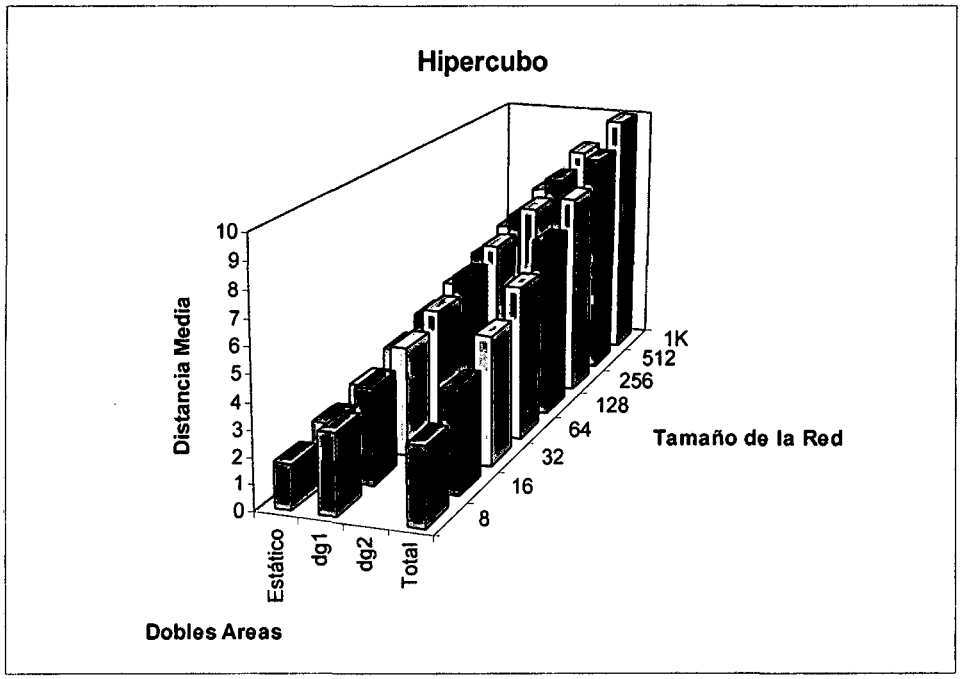


Figura 5-13 Áreas de Gravedad Dobles. Hipercubo

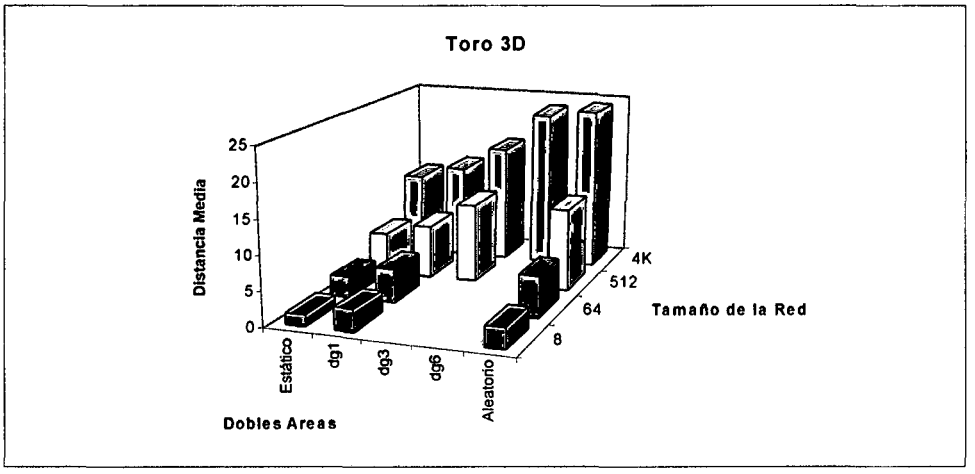


Figura 5-14 Áreas de Gravedad Dobles. Toro 3D

Hasta aquí se han mostrado las tendencias de los supernodos para todas las topologías. Como se puede observar en las gráficas anteriores, se han encontrado resultados similares para todas las topologías con diferentes tamaños de red que muestran la escalabilidad de los métodos de construcción de supernodos.

5.3.3 Escalabilidad de los métodos DRB

Este punto quiere mostrar la escalabilidad de los métodos DRB, es decir, la proporción que existe entre el alargamiento del camino y el tamaño de la red. El objetivo esperado, que mostraría una regularidad de DRB frente al tamaño de la red, es que exista una proporción constante entre un valor y el otro.

En las siguientes figuras se observa la escalabilidad de los métodos. Para ello se han representado gráficamente los datos anteriores solo para dos supernodos y para todos los tamaños de red. En el caso del hipercubo, sólo se muestra en la gráfica un supernodo, que son hipercubos de tamaños menores. El encaminamiento estático y el aleatorio se han incluido por propósitos comparativos. En las Figura 5-15, Figura 5-16, Figura 5-17 y Figura 5-18 se demuestra que los métodos de supernodos se mantienen en un punto intermedio entre ellos.

Las Figura 5-15 y Figura 5-16, muestran la distancia promedio utilizando los métodos de supernodos de subtopologías para toros 2D y *"midimews"* con diferentes tamaños de red desde 9 hasta 64K nodos. Se muestran dos métodos de supernodos: una fila de tamaño n y una columna de tamaño m para redes de n x m nodos. Cuando $m = n$ los resultados son los mismos para ambos.

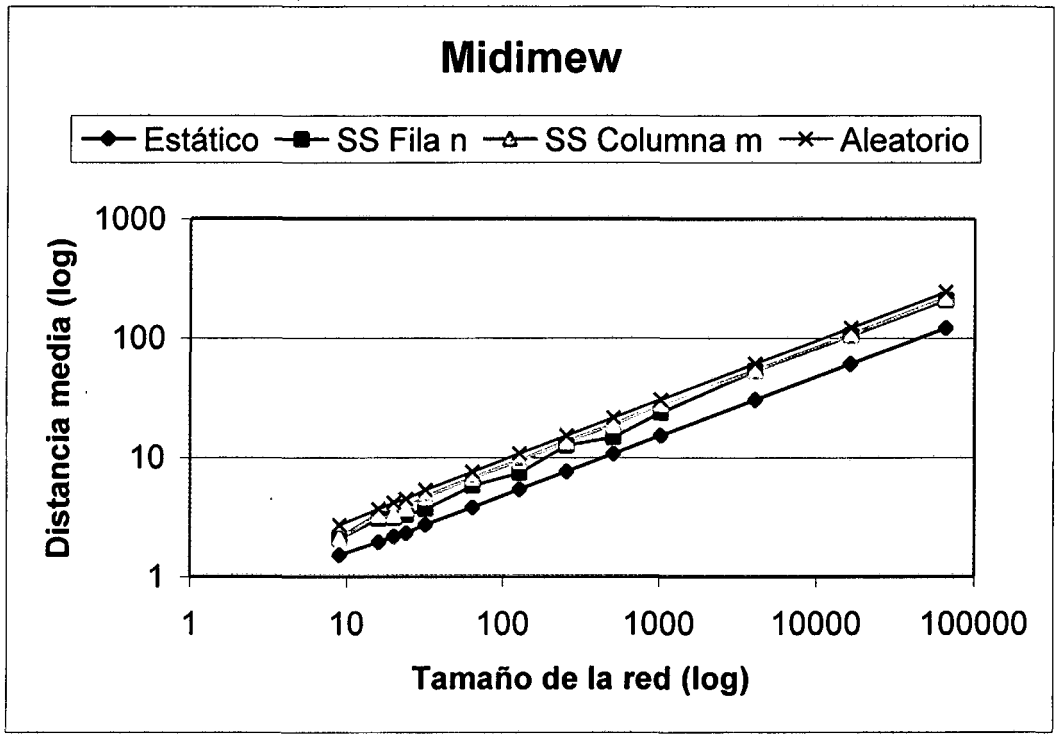


Figura 5-15 Escalabilidad de los Supernodos: "*Midimew*"

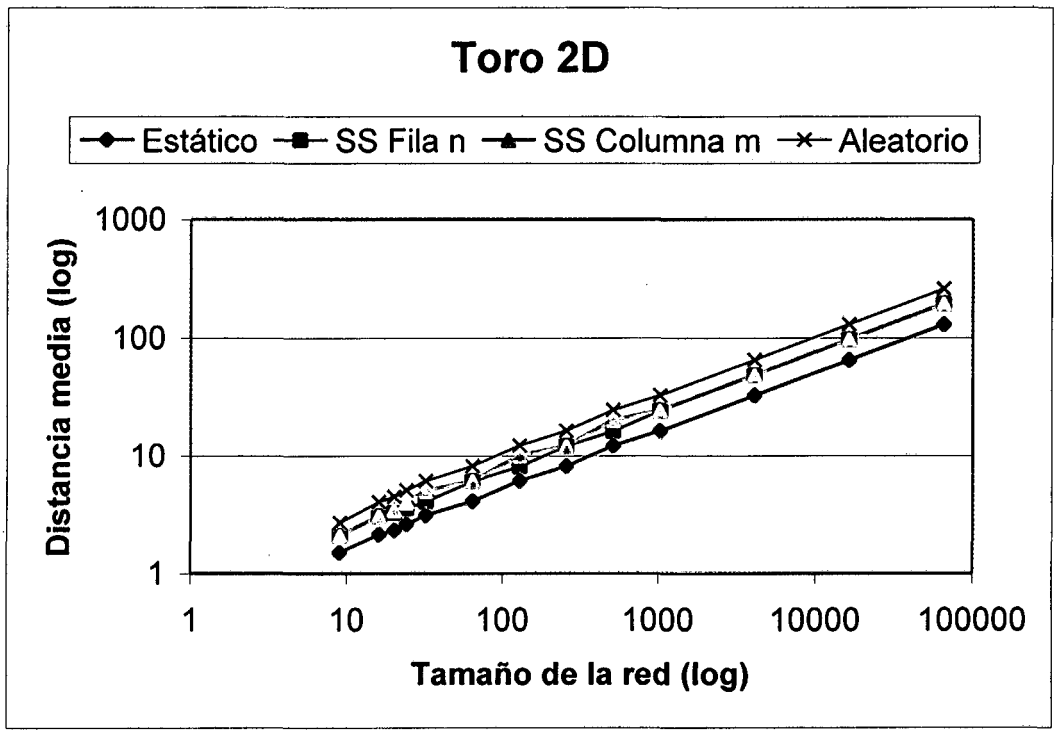


Figura 5-16 Escalabilidad de los Supernodos: Toro 2D

En las Figura 5-17 y Figura 5-18 se muestran toros 3D e hipercubos de 8 a 64K. En la figura 34 se muestran toros 3D con dos métodos: una fila de tamaño *n* y una subtopología *n* x *m*. En la figura 35 se observa un supernodo igual a la mitad de la dimensión para topologías hipercubos.

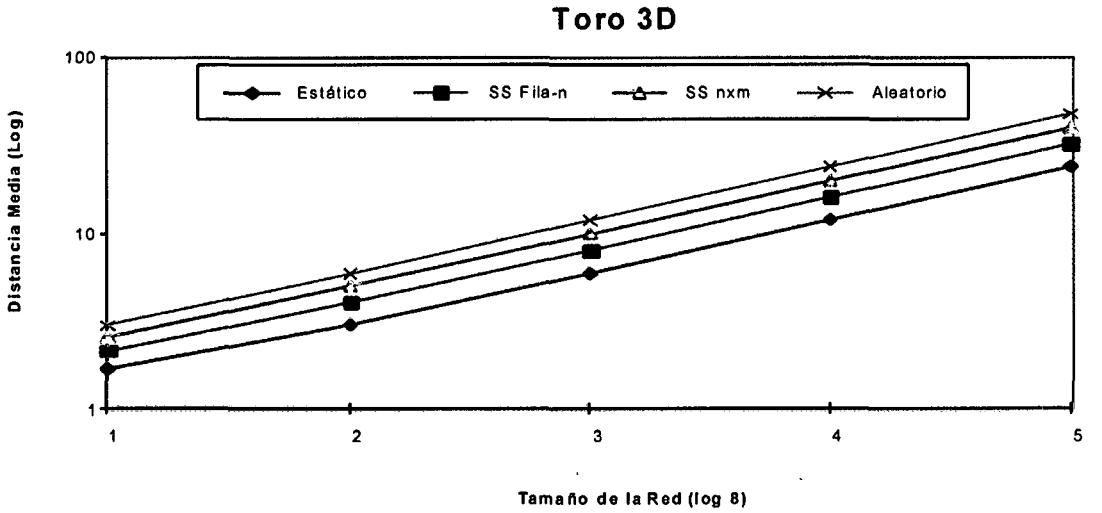


Figura 5-17 Escalabilidad de los Supernodos: Toro 3D

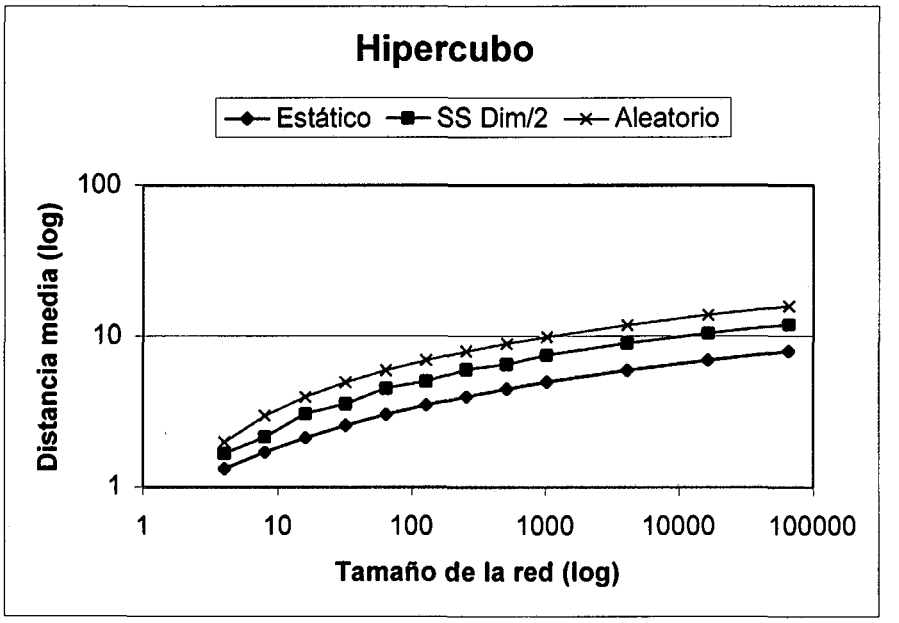


Figura 5-18 Escalabilidad de los Supernodos: Hipercubo

Para todas estas gráficas (Figura 5-15, Figura 5-16, Figura 5-17 y Figura 5-18), los métodos DRB crecen de manera proporcional con el tamaño de la red y se mantienen entre los casos estático y aleatorio, lo cual demuestra la escalabilidad de DRB frente al tamaño de la red. Este es un resultado muy importante porque implica

que un cierto resultado conseguido con DRB para una cierta aplicación con un determinado patrón de comunicaciones puede mantenerse de manera similar al escalar la red.

5.4 Conclusiones

En conclusión, en este capítulo se ha presentado la experimentación completa de la primera componente de DRB. También se han analizado al nivel de tablas y gráficos los resultados obtenidos. Se ha podido observar que se ha creado una metodología sistemática para la definición de los supernodos para cada topología. Estas definiciones comunes implican una conducta similar para supernodos equivalentes en diferentes topologías.

Con esta experimentación, pues, se conocen cuáles son las capacidades potenciales de DRB. Es decir, para una cierta topología y método de definición de supernodos, se conoce hasta cuánto es posible "abrir" los caminos, es decir, cuántos caminos alternativos o qué ancho de banda máximo se puede conseguir. Asimismo, se conoce el precio pagado en alargamiento del camino que supone cada alternativa. Se observa que el número de caminos alternativos es proporcional al tamaño de los supernodos y al alargamiento, con lo que las ganancias conseguidas se mantienen al escalar los supernodos o la topología. Este hecho informa de la generalidad de DRB para un amplio rango de casos como los analizados.

Con estos datos, dadas una red de interconexión concreta y una aplicación a ejecutar sobre ella, es posible conocer el ancho de banda y la latencia base que cada uno de las posibles combinaciones de supernodos y metacaminos del método DRB aplicado sobre ese caso es capaz de ofrecer. Con esta experimentación, hemos obtenido una información sobre la red para utilizarla en DRB cuando, de manera dinámica, necesite expandir los caminos.

A partir de ahora, que ya se conocen los límites teóricos máximos de DRB, hace falta ver cómo se utilizan cuando hay tráfico real. Es decir, hay que evaluar la política de encaminamiento bajo tráfico real presentada en el capítulo anterior. En este sentido, es necesario analizar cómo se configuran y eligen los metacaminos y los caminos multipaso para mantener la latencia de comunicaciones bajo ciertos límites. Esta evaluación se ha realizado mediante simulación. A este cometido se dedica el siguiente capítulo que presenta la experimentación realizada con la herramienta de simulación funcional construida para este fin.

Capítulo 6 Evaluación de DRB: Rendimiento dinámico

6.1 Introducción

En este capítulo se presenta la experimentación realizada sobre el método DRB: el rendimiento que presenta frente a tráfico real. En el capítulo 5 se han presentado los experimentos que muestran el incremento en ancho de banda que puede ofrecer DRB a las aplicaciones, según se utilizase un conjunto u otro de metacaminos, frente al alargamiento del camino que supone esta técnica. En este capítulo, se miden las prestaciones dinámicas de DRB en presencia de una carga real de mensajes en la red de interconexión. Para realizar esta evaluación se ha llevado a cabo una experimentación basada en la simulación, utilizando el simulador funcional presentado en el capítulo tres como herramienta de prueba.

Para realizar la experimentación hace falta tener un sistema de representar la carga real de las aplicaciones paralelas. De esta carga, lo que nos interesa a nosotros es el tráfico de mensajes generado por las aplicaciones. Este tráfico se puede obtener de aplicaciones reales o de "*benchmarks*", que son un conjunto de casos de prueba específicamente seleccionados. En nuestro caso hemos elegido un conjunto de "*benchmarks*" porque tienen una serie de ventajas. Primero, porque son más representativos al ser capaces de abarcar un mayor amplio rango de situaciones que un conjunto arbitrario de aplicaciones y, segundo, porque son parametrizables y de este modo se puede tratar de medir cuestiones específicas. En nuestro caso, con los "*benchmarks*" que hemos utilizado seleccionamos un conjunto de patrones de comunicación que aparecen en los programas reales de las aplicaciones científicas y técnicas paralelas más comunes. Estos patrones tienen la propiedad de utilizar la red de interconexión de manera extensiva, con lo cual, por un lado, podemos observar el "efecto" colectivo de DRB y, por el otro, podemos distribuir la carga general de la red con lo cual DRB tendrá que trabajar en presencia de carga existente.

La experimentación se ha realizado de manera comparativa evaluando otros métodos de encaminamiento existentes en la literatura. Estos métodos son el encaminamiento mínimo estático y el encaminamiento completamente adaptativo de camino mínimo. El encaminamiento estático se ha elegido como método base de referencia que nos sirve como valor "umbral" conseguido con el método más simple, es decir, las prestaciones de la red "desnuda" sin incluir ningún método de mejora que suponga ningún coste extra. El encaminamiento completamente adaptativo se ha elegido por permitir todos los caminos mínimos entre fuente y destino y es uno de los que presentan mejores prestaciones [41]. El encaminamiento completamente adaptativo representa el "*state-of-the-art*" en cuanto a métodos de encaminamiento y es la referencia obligada a comparar con DRB, ya que éste último presenta características adaptivas.

La experimentación presentada en este capítulo está estructurada de la siguiente manera. Primero, se evalúan tres aspectos específicos sobre el mensaje de reconocimiento que genera DRB con la información de latencia y que viaja desde el nodo destino al nodo fuente. Estos tres aspectos son el "*overhead*" que supone el mensaje de reconocimiento de DRB en la red, la influencia de la generación temprana

del mensaje de reconocimiento (es decir, generado en cuanto se detecta un incremento de latencia del mensaje en lugar de generarlo al llegar al nodo destino) y la influencia del retardo del mensaje de reconocimiento con la información de latencia, en su viaje hacia el nodo fuente. Con esta evaluación se quiere conocer la respuesta transitoria de DRB, el tiempo de respuesta de DRB y la robustez de DRB frente a la información de latencia que DRB necesita para configurar los metacaminos y seleccionar los caminos multipaso. Esta experimentación nos servirá, además, para configurar el resto de experimentación de tipo más general y exhaustivo, donde se evalúa DRB para un conjunto de redes de interconexión de diversos tamaños (toros e hipercubos) y de patrones de comunicación.

La experimentación siguiente se enfoca en dos puntos principales. El primero es la respuesta en latencia con patrones de comunicación persistentes tomados de aplicaciones numéricas (*"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"*) y el segundo, es la respuesta respecto a un patrón de comunicaciones que provoca la aparición de un "*hot-spot*" en la red. Para todo ello se evalúa la respuesta en latencia, el "*throughput*" de mensajes y la desviación estándar de la latencia. Otros aspectos asimismo evaluados son la influencia de la longitud del mensaje y la escalabilidad de DRB respecto el aumento del tamaño de la red de interconexión.

Toda esta experimentación se ha realizado utilizando el simulador funcional de redes de interconexión presentado en el capítulo 3. Para ello, este simulador incorpora todos los aspectos de funcionamiento del encaminador DRB descrito en el capítulo 4, así como, la funcionalidad del encaminamiento estático y el adaptivo de caminos mínimos.

Los siguientes apartados presentan, en una serie de subapartados, cada uno de los aspectos mencionados en el párrafo anterior.

6.2 Evaluación de las prestaciones de DRB

En este punto, mostramos los aspectos generales que definen la experimentación. Estos aspectos son la definición y caracterización de la carga, las características de la red de interconexión, las herramientas utilizadas para evaluar, la metodología de trabajo, la obtención de resultados, su procesamiento y representación gráfica. Para realizar la experimentación se han elegido una serie de parámetros que resultan representativos del

entorno en el que se centran las redes de interconexión como son la topología, el tamaño de la red, el patrón de comunicaciones y el tamaño del mensaje.

Como se explica en [40], el conjunto de pruebas elegido debe ser suficientemente representativo de la situación a simular. En sus propias palabras, "el conjunto de pruebas a realizar representa un espacio de programa que debe presentar unas propiedades adecuadas para que la evaluación sea descriptiva". Las principales propiedades mencionadas en esa referencia son el tamaño, la densidad y la granularidad del conjunto de pruebas y el alcance del espacio de programa. Las simulaciones tratan de representar el comportamiento del mundo real mediante el envío de paquetes a través de los enlaces de la red de acuerdo a un patrón de comunicaciones específico. Tal y como se comenta en [109], un conjunto de patrones de comunicación bajo un espectro de carga de la red suficientemente amplio es un buen conjunto de pruebas de las redes de interconexión que se acerca a la evaluación realizada con aplicaciones reales.

Para la evaluación del caso que nos ocupa, debemos seleccionar dos elementos. Por un lado, una plataforma de prueba, que será la red de interconexión sobre la que se pruebe el método. Por el otro, una carga de comunicaciones que sea representativa de lo que son las condiciones de carga de la red. A continuación, se precisan cada uno de los aspectos mencionados que definen la experimentación.

6.2.1 Redes de interconexión

En nuestra experimentación, hemos utilizado un conjunto amplio de diferentes redes de interconexión directas de las presentadas en el capítulo 2. Sin embargo, para nuestro caso concreto, hemos elegido mostrar de manera comparativa el método de encaminamiento DRB para un conjunto de topologías tipo n-cubos k-arios, concretamente de diversos tamaños, toros e hipercubos, como conjunto de redes de interconexión. Estas redes se han elegido por ser las más utilizadas en la literatura tanto como objeto de estudio como para implementación de máquinas reales. De este modo, la mayoría de estudios de métodos de encaminamiento se evalúan sobre n-cubos k-arios, de manera que, para que la información presentada en este capítulo se pueda comparar con otras experimentaciones de la literatura, nosotros nos centramos en presentar la evaluación realizada con este tipo de redes. Como técnica de control del flujo, hemos utilizado la técnica "*wormhole*", descrita en el capítulo 2, que es la que obtiene mejores prestaciones frente a un espacio de almacenamiento mínimo.

6.2.2 Carga de comunicaciones

Como carga de comunicaciones, se ha elegido una serie de configuraciones de pares de nodos fuente y destino que se envían mensajes representados por patrones de comunicación. La carga de comunicaciones se representa por estos patrones, que definen entre que nodos se envían mensajes, con una longitud de los mensajes y con una frecuencia de generación de mensajes, que significa, como se vio en el capítulo 3, el tiempo medio de ejecución entre el envío de dos mensajes. Esta generación de mensajes se implementa mediante una función de distribución de probabilidades exponencial cuya media es el tiempo entre la llegada de dos mensajes. Como se explicó en el capítulo 3, esta función expresa el comportamiento de las tareas individuales del programa paralelo que realiza acciones de cómputo de duración independiente entre ellas y al finalizar cada acción de cómputo se realiza una acción de comunicación.

Los experimentos se han realizado para un rango de carga del tráfico de comunicación que va desde baja carga hasta la saturación. Se ha elegido un tamaño de paquete de 10 "*flits*" por ser un valor representativo del tamaño de los paquetes de los encaminadores reales, como se vio en el capítulo 2, y por ser un valor común elegido en la literatura para evaluar redes de interconexión. Para una longitud del paquete de 10 "*flits*", y dado que el simulador hace avanzar un "*flit*" en un ciclo de simulación, se puede considerar **baja carga** un *intervalo entre mensajes de 120 ciclos* de simulación, lo que es menor de un 10% de la capacidad de un enlace. La *saturación* supone una inyección continua de paquetes, por lo que se le asigna un *intervalo entre mensajes de 10 ciclos*. Los experimentos se han evaluado para un conjunto de valores de carga entre ambos límites (10 y 120 ciclos entre mensajes) con un paso de 10 ciclos entre ellos, es decir, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 y 120 ciclos/mensaje.

Esta tasa de mensajes generados se llama *carga aplicada* ("*Applied Load*"). La inversa de este valor, ciclos entre mensajes, (ciclos/mensaje), nos da la frecuencia de generación de mensajes. Pero dado que, cuando se llega a tasas altas, la red llega a la saturación y no se absorbe toda la carga, se produce el rechazo de parte de ella. La parte de la carga realmente inyectada que circula por la red, se llama *carga aceptada* ("*Accepted Load*"), y que, cuando existe rechazo, es menor que la carga aplicada. En los resultados no se presenta la carga aplicada, sino la aceptada, ya que este último es el valor que realmente esta circulando por la red, y es el que nos interesa medir.

6.2.3 Patrones

Como se ha comentado en la introducción, se ha seleccionado una serie de "*benchmarks*" sintéticos para evaluar DRB. Las ventajas de utilizar este tipo de

"*benchmarks*" frente a aplicaciones reales son que se tiene un mayor control de lo que éstos representan y se puede asegurar que cubren todo el espectro de aplicaciones posible. Además, son parametrizables y se puede sintonizar de manera muy precisa los aspectos que se quieren medir. Estos "*benchmarks*" se componen de una serie de patrones de comunicación. Los patrones elegidos se pueden englobar en dos categorías: patrones sistemáticos y patrones específicos. Los patrones sistemáticos se forman de una manera metodológica atendiendo a un esquema algorítmico mientras que los específicos se configuran "a mano" para representar situaciones especiales.

El primer conjunto de patrones elegidos, los patrones sistemáticos, representa las computaciones numéricas que se realizan en muchos programas de cálculo matemático como la transformada rápida de Fourier y la convolución, por ejemplo, que se utilizan en el cómputo científico y técnico [41]. Estos patrones son: *"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"*. Todos estos patrones definen una permutación entre todos los nodos de la red de manera que cada nodo envía a algún otro nodo, por lo que pueden convertirse en patrones que fuerzan a la red a trabajar en situaciones límite. El nodo destino de cada nodo fuente esta formado por una transformación de los dígitos del número del nodo expresado en binario. A continuación, se incluyen las expresiones matemáticas de las transformaciones que realiza cada uno de los patrones para determinar el nodo destino a partir del nodo fuente [71]. Sea el nodo fuente formado por *n* coordenadas $\{ a_{n-1}, a_{n-2}, ..., a_1, a_0\}$.

- ✓ El patrón *"Butterfly"* se forma intercambiando los bits más y menos significativos: el nodo con coordenadas binarias $a_{n-1}, a_{n-2}, ..., a_1, a_0$ se comunica con el nodo Destino(*Butterfly*)= $\{a_0, a_{n-2}, ..., a_1, a_{n-1}\}$

- ✓ Bajo el patrón *"Bit-Reversal"* el nodo con coordenadas binarias $a_{n-1}, a_{n-2}, ..., a_1, a_0$ se comunica con el nodo Destino(*Bit-Reversal*)= $\{a_0, a_1, ..., a_{n-2}, a_{n-1}\}$.

- ✓ El patrón *"Perfect Shuffle"* rota un bit a la izquierda: el nodo con coordenadas binarias $a_{n-1}, a_{n-2}, ..., a_1, a_0$ se comunica con el nodo Destino(*Perfect Shuffle*)= $\{a_{n-2}, a_{n-3}, ..., a_0, a_{n-1}\}$.

- ✓ En el patrón *"Matrix Transpose"* el nodo con coordenadas binarias $a_{n-1}, a_{n-2},..., a_1, a_0$ se comunica con el nodo Destino(*Matrix Transpose*) = $\{a_{n/2-1},...,a_0, a_{n-1},...,a_{n/2}\}$.

Como patrón específico se ha elegido el patrón de "*hot-spot*" que se ha presentado en el ejemplo básico del capítulo 4. En este patrón, un conjunto de canales que no comparten ni el origen ni el destino, comparten una fragmento del camino que recorren

donde se produce una gran concentración de canales (**Figura 6-1**). Este es un patrón de concentración de paquetes en una zona localizada de la red. Este patrón se evalúa sobre una red toroidal de 8x8 nodos.

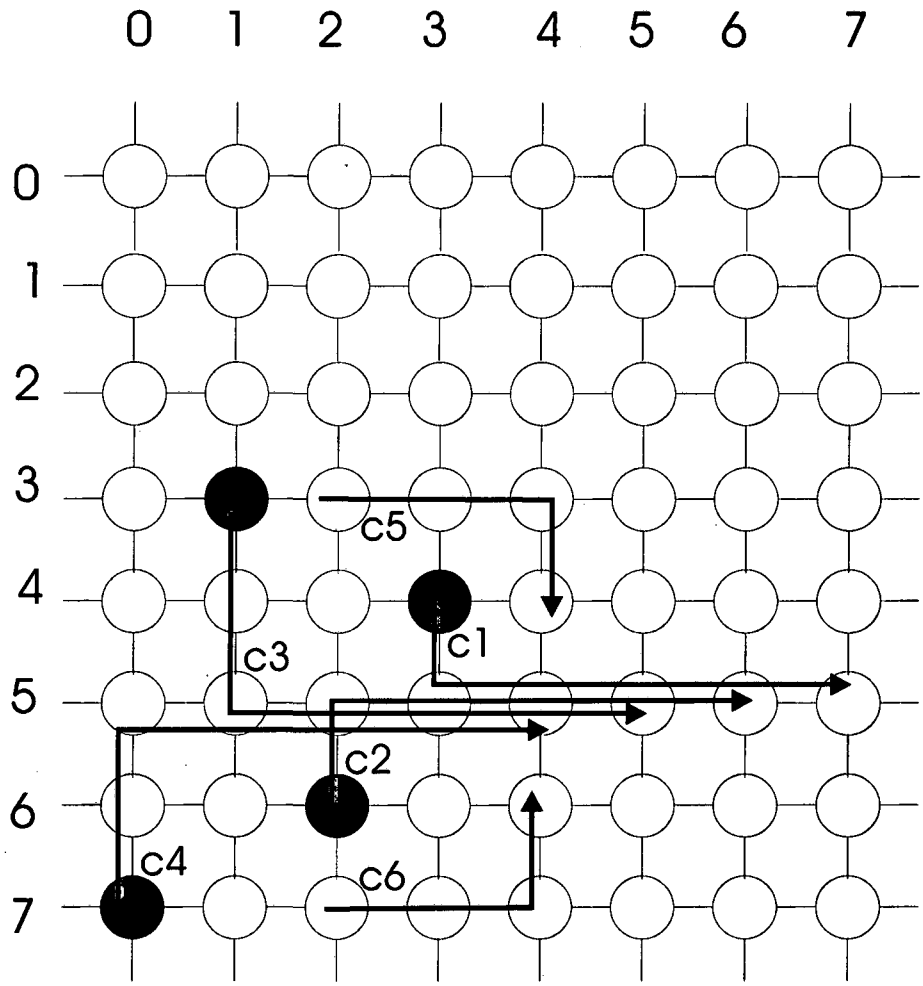


Figura 6-1 Patrón de "hot-spot"

6.2.4 Métodos de encaminamiento

Los algoritmos de encaminamiento utilizados en la experimentación son el DRB, el estático mínimo y el adaptativo completo de caminos mínimos. Para el tipo de topologías utilizadas en la experimentación, el encaminamiento estático mínimo es el algoritmo clásico de encaminamiento por dimensiones (DOR) que encamina los mensajes por orden creciente de dimensión.

Respecto al encaminamiento adaptativo, este algoritmo elige, para encaminar un paquete en cada uno de los encaminadores que atraviesa, el primer enlace de salida disponible del conjunto de enlaces que lo acercan al destino. Si el enlace siguiendo encaminamiento DOR está libre, lo elige; en caso contrario consulta si algún otro enlace que proporcione un camino mínimo esta libre con objeto de encaminar el paquete a

través de él. En el caso de que estén todos ocupados, bloquea el mensaje que se espera hasta que algún enlace de camino mínimo se libere y lo envía a través de él.

Respecto al encaminamiento DRB, ya se ha explicado en capítulos anteriores su funcionamiento. Para realizar la experimentación, se ha configurado, en cada caso, un conjunto de supernodos y de metacaminos que buscasen los caminos alternativos que representen una menor utilización. Concretamente, se han utilizado metacaminos compuestos por hasta tres caminos multipaso, que incluye el camino original DOR más dos caminos alternativos.

6.2.5 Metodología de trabajo

La experimentación realizada se presenta en varias partes. Primero, se evalúan tres aspectos específicos sobre el mensaje de reconocimiento que genera DRB con la información de latencia y que viaja desde el nodo destino al nodo fuente. Estos tres aspectos son el "*overhead*" que supone el mensaje de reconocimiento de DRB en la red, la influencia de la generación temprana del mensaje de reconocimiento (es decir, generado en cuanto se detecta un incremento de latencia del mensaje en lugar de generarlo al llegar al nodo destino) y la influencia del retardo del mensaje de reconocimiento con la información de latencia, en su viaje hacia el nodo fuente. El primero de estos aspectos se ha evaluado para todos los patrones presentados, ya que los valores son diferentes e interesa conocer su influencia en cada patrón. Para los otros dos aspectos se presenta sólo los resultados para el patrón de "hot-spot", ya que los resultados para estos casos son independientes del patrón de comunicación y sólo dependen de la carga de tráfico.

Con esta evaluación se quiere conocer la respuesta transitoria de DRB, el tiempo de respuesta de DRB y la robustez de DRB frente a la información de latencia que DRB necesita para configurar los metacaminos y seleccionar los caminos multipaso. Esta experimentación nos servirá, además, para configurar el resto de experimentación de tipo más general y exhaustivo, donde se evalúa DRB para un conjunto de redes de interconexión de diversos tamaños (toros e hipercubos) y de patrones de comunicación.

La experimentación siguiente es una experimentación exhaustiva de DRB frente a patrones y topologías y se enfoca en dos puntos principales. El primero es la respuesta en latencia con patrones de comunicación persistentes tomados de aplicaciones numéricas (*"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"*) y el segundo, es la respuesta respecto a un patrón de comunicaciones que provoca la aparición de un "*hot-spot*" en la red. Para todo ello se evalúa la respuesta en latencia, el "*throughput*" de mensajes y la desviación estándar de la latencia. Otros aspectos

asimismo evaluados son la influencia de la longitud del mensaje y la escalabilidad de DRB respecto el aumento del tamaño de la red de interconexión.

Los experimentos fueron ejecutados varias veces con diferentes semillas y promediados para ser consistentes. La simulación fue ejecutada para un millón de paquetes como media y los efectos de los primeros 50.000 paquetes entregados no se tiene en cuenta en los resultados con objeto de eliminar los efectos transitorios en las simulaciones.

6.2.6 Resultados medidos

Como resultados de los experimentos, hemos medido tres aspectos: Primero, la "latencia media" de las comunicaciones de la red de interconexión, segundo, el "*throughput*" medio y, tercero, la "distribución de la carga" de tráfico en la red.

La "latencia de comunicación" se mide como el tiempo total transcurrido desde que el mensaje viaja desde el nodo fuente hasta el nodo destino, incluyendo el tiempo que el mensaje espera por ser inyectado en la red. Para un conjunto de valores de latencia Xi medidos, la "latencia media" $\overline{X}$ se calcula promediando todas las latencias de todos los mensajes enviados y se mide en ciclos de simulación:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

También se obtiene la desviación estándar del valor medio de la latencia (*StdDevLat*) y los valores máximos de latencia. La desviación estándar es la raíz cuadrada de la varianza de la muestra S^2 cuya expresión se muestra a continuación. La varianza mide la dispersión media de los valores de una muestra respecto a su valor medio.

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

El "*throughput*" se mide como la relación porcentual entre la carga aceptada (cantidad de información entregada) y la carga de comunicación aplicada (tasa de inyección). Ambas cargas de comunicación se miden como el número de mensajes por unidad de tiempo.

Finalmente, con objeto de mostrar la "distribución de la carga" de los mensajes en la red, se calcula la *latencia media en cada enlace* de la red. Esta latencia media de un enlace se calcula promediando cada uno de los tiempos que cada mensaje tuvo que esperar en ese enlace cuando lo atravesó. Por lo tanto, es una media por enlace y no por mensaje. Esta información se muestra para la experimentación realizada con el patrón de "*hot-spot*".

A modo de ejemplo, se incluye una de las gráficas (Figura 6-2) donde se muestran los elementos que aparecen en todas ellas. El eje Y muestra la magnitud medida en cada caso como es la latencia, la desviación estándar de la latencia y el "*throughput*". El eje X muestra la carga aceptada (*AccLoad/C.Ace*) medida en ciclos/mensaje. Esta carga se genera a partir del rango de carga aplicada (*AppLoad/C.Apli*), descrito anteriormente. El rango de carga aceptada varía en función de la topología y el patrón de comunicaciones. En el análisis de la información que nos proporciona esta figura se observa que al ir incrementando la carga aplicada aumenta la carga aceptada, hasta llegar a un valor en que la carga aceptada, en lugar de incrementarse, se decrementa al incrementar la carga aplicada. Esto es debido a que la red está en la zona de saturación y el hecho de inyectar más mensajes provoca mayores colisiones y mayores rechazos de mensajes.

Al mismo tiempo que estas latencias se reducen, el "*throughput*" conseguido se mejora. Es decir, el número de mensajes entregados con menor latencia se incrementa. Este aspecto se observa en las gráficas presentadas, en las que en el eje x se presenta la carga aceptada, que es una función de la carga aplicada. Se observa que cada punto de DRB representa una carga mayor que el correspondiente punto de los otros métodos de encaminamiento porque se sitúa más a la derecha en el gráfico. Los otros métodos se comportan peor porque, con ellos, la red se satura antes y, por tanto, consiguen valores menores de carga aceptada.

El diferente comportamiento de los tres métodos se pone de manifiesto si tenemos en cuenta que los puntos (A), (B) y (C) mostrados en la Figura 6-2 corresponden a los de carga aplicada máxima (C.Apli=10) para todos ellos. Por tanto, serían puntos equivalentes, y, sin embargo, vemos las diferencias existentes en la carga aceptada (C.Ace) y el valor de "*Latency*".

Las leyendas se muestran en inglés ya que son producidas por el simulador funcional que es una herramienta que, en aras de facilitar su uso y difusión, se encuentra, en su versión actual, en versión inglesa.

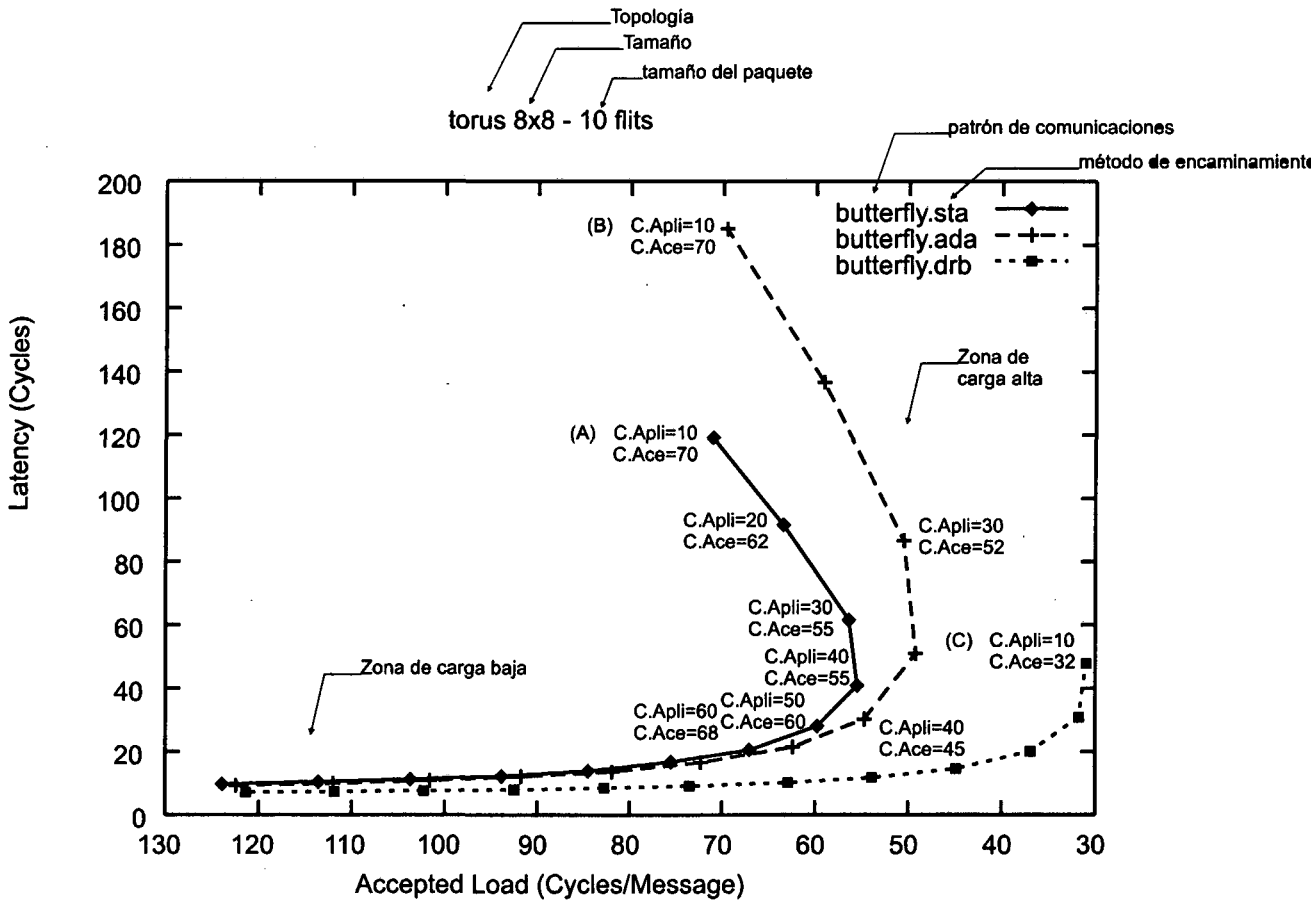


Figura 6-2 Gráfica de ejemplo de los resultados obtenidos de latencias

Como se ha comentado, además de los valores de latencia, también se han medido las desviaciones estándar de esa latencia. Se presentan las gráficas en cada caso que incluyen esta información. A modo de ejemplo, la Figura 6-3 muestra una gráfica de desviaciones de las latencias de la gráfica de latencia anterior. En este caso, se representa los valores obtenidos de las desviaciones estándar de las latencias medidas en ciclos de simulación para cada uno de los valores de la carga aplicada ("*Applied Load*"), tal y como se ha definido. Con estas gráficas, se pretende informar de la uniformidad conseguida por los métodos de encaminamiento. En este sentido, se considerará mejor un método que, para un valor de carga aplicada, presente una desviación de las latencias menor, siempre que el valor asociado de latencia sea menor, por supuesto. Esto significa que las latencias, a parte de su valor absoluto, presentan una variación menor. Como se recordará, el objetivo de latencia uniforme es uno de los principales objetivos perseguidos por DRB, ya que permite su predicción y con ello, la realización de una asignación efectiva de tareas a nodos de cómputo. Como se observa en la Figura 6-3, DRB presenta, en la mayoría de los casos, desviaciones de las latencias menores que los otros métodos de encaminamiento, sobre todo a valores de carga aplicada altos, que es cuando los valores de latencia absoluta se disparan. Como se observa en la gráfica, las desviaciones conseguidas con DRB, muestran un suave crecimiento a medida que se

aumenta la carga, mientras que para el caso del adaptivo son mayores y en el caso de carga muy alta (10 ciclos entre mensajes) decrece la desviación porque la saturación de la red hace que la mayoría de los mensajes sufran unos valores de latencia muy altos similares. Lo mismo ocurre en esta gráfica para el encaminamiento estático, donde la desviación crece a medida que aumenta la carga hasta un valor (30 ciclos entre mensajes) que decrece por la misma razón que el caso adaptivo: la red está saturada y la latencia toma valores muy altos para todos los mensajes.

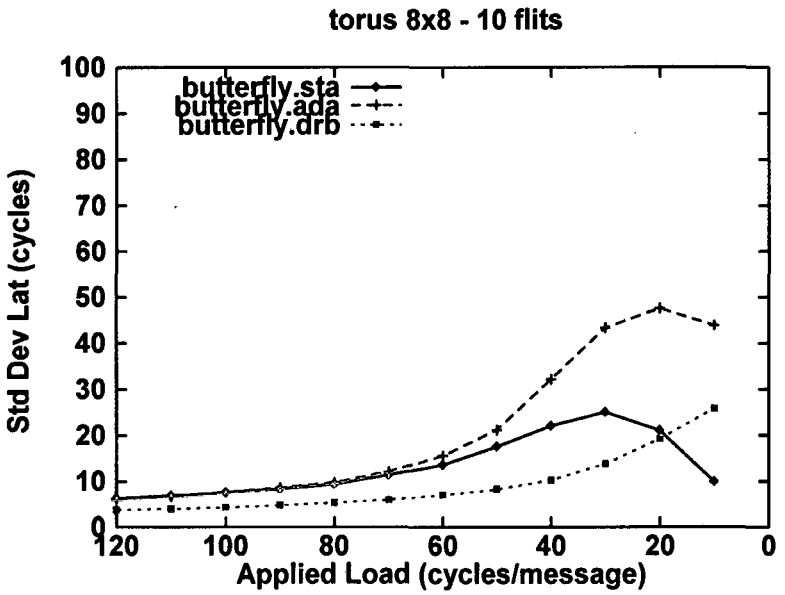


Figura 6-3 Gráfica de ejemplo de los resultados obtenidos de desviación estándar de las latencias

A continuación, se presentan los experimentos realizados para cada caso y se analizan los resultados obtenidos.

6.3 Influencia del mensaje de reconocimiento

En este punto se quiere mostrar la influencia del mensaje de reconocimiento de DRB. Evidentemente, el mensaje de reconocimiento enviado hacia los nodos origen para informar de la latencia existente en la red supone un incremento de la carga de comunicaciones. Analizando con profundidad este aspecto, se puede observar que los mensajes de reconocimiento viajan por caminos diferentes a los mensajes que los originan y por tanto no tienen influencia sobre ellos, sino sobre otros mensajes de la red. Otra observación es que estos mensajes de reconocimiento son de longitud muy corta al ser solo mensajes de control, sin datos de usuario.

Hemos realizado una experimentación en la que incluimos el "overhead" del mensaje de reconocimiento sobre la red de interconexión y otra en la que el mensaje de

reconocimiento no supone ningún "overhead" sobre los demás mensajes que viajan por la red de interconexión. En la primera experimentación, que hemos llamado DRB-ACK, el mensaje de reconocimiento se inyecta y viaja como un mensaje más en la red de interconexión. En la segunda, llamada DRB, el mensaje de reconocimiento llega de manera automática al origen para entregar su información de latencia al módulo configurador de los metacaminos sin consumir ancho de banda de la red de interconexión. Esta situación correspondería al caso en el que se dispusiera de una red específica para enviar los mensajes de reconocimiento, o que el sistema de comunicaciones incluyera por defecto un mensaje de reconocimiento, el cual aprovecharía DRB. El simulador funcional presentado en el capítulo 3 utilizado para esta experimentación incluye ambas posibilidades de manera seleccionable por el usuario.

Hemos elegido una serie de casos representativos para evaluar este aspecto. Estos casos son los patrones sistemáticos sobre las topologías de hipercubo y toro y el patrón de "*hot-spot*". La Tabla 6-1 muestra la experimentación realizada para mostrar la influencia del mensaje de reconocimiento.

Topología	**Tamaño (Nodos)**	**Patrones**	**Encaminamiento**	**Figura**
Toro	4x4 (16)	Butterfly Reversal Shuffle Transpose	DRB DRB-ACK	Figura 6-4
Hipercubo	4D (16)			Figura 6-5
Toro	8x8 (64)	Hot-spot		Figura 6-6

Tabla 6-1 Experimentación para mostrar la influencia del mensaje de reconocimiento

La Figura 6-4 (a) muestra los resultados para el toro 2D de 16 nodos donde se observa que el DRB con mensaje de reconocimiento empeora su situación respecto al DRB ideal. Para el patrón *"Butterfly"*, la pérdida en latencia es mínima, sólo siendo significativa a cargas altas. La desviación estándar (Figura 6-4 (b))es un poco mayor para el caso que incluye el mensaje de reconocimiento debido al retraso del mismo en llegar al nodo.

Para los patrones de *"Bit-Reversal"* y *"Matrix Transpose"* la pérdida en latencia es constante para toda la carga suponiendo acerca de un 30% o menor respecto el DRB ideal. Esta pérdida es debida a que los mensajes de reconocimiento colisionan con los caminos de otros mensajes generados por el propio patrón. La desviación para estos dos patrones se incrementa para el caso de ACK a cargas altas debido a la razón comentada para el patrón *"Butterfly"*. Para el patrón *"Perfect Shuffle"* la pérdida es nula debido a que los caminos que recorren los mensajes de reconocimiento no colisionan con mensajes en la red de interconexión. La desviación estándar también es la misma en ambos casos.

La Figura 6-5 (a) muestra los resultados para el caso de hipercubo. Se observan resultados similares para todos los casos. En el caso de la topología hipercubo para todos los patrones, la pérdida es nula porque los mensajes de reconocimiento no utilizan los enlaces que usan los mensajes de los canales de usuario. La desviación estándar (Figura 6-5 (b)) también presenta resultados casi idénticos para todos los patrones y valores de carga. Esto demuestra que, para la topología hipercubo, no tiene influencia la inclusión del mensaje de reconocimiento debido a la configuración de enlaces de la red.

La Figura 6-6 (a) muestra los resultados para el patrón de "*hot-spot*". Se observan resultados similares como en los casos anteriores. En el caso del patrón de "*hot-spot*", la pérdida es nula porque los mensajes de reconocimiento no utilizan los enlaces que usan los mensajes de los canales de usuario. La desviación estándar (Figura 6-6 (b)) también presenta resultados casi idénticos excepto en el último punto de carga en el que debido a la saturación en la red, se produce mayor variación en los resultados para el caso en el que no se incluye la influencia del mensaje de reconocimiento.

Con esta experimentación, hemos mostrado los dos casos extremos de DRB. Cuando no se genera mensaje de reconocimiento y la información se comunica directamente al nodo origen y cuando sí se genera y éste viaja sin prioridad por la misma red de datos. Soluciones intermedias, como la inclusión de una red de control compuesta por una línea serie o por una red completa, o la asignación de prioridades a los mensajes de reconocimiento, ofrecerían un rango de prestaciones entre las dos soluciones presentadas. La reserva de un pequeño porcentaje del ancho de banda de la red para los mensajes de ACK, eliminaría esas diferencias ya que ambos tipos de mensaje no colisionarían. Esta solución se podrá ver más adelante que es interesante pues al permitir DRB aprovechar un ancho de banda mayor ("*throughput*" mayor) de la red, puede dedicar una pequeña fracción para los mensajes de ACK, con lo que no existiría una penalización real por el hecho de monitorizar la red de interconexión.

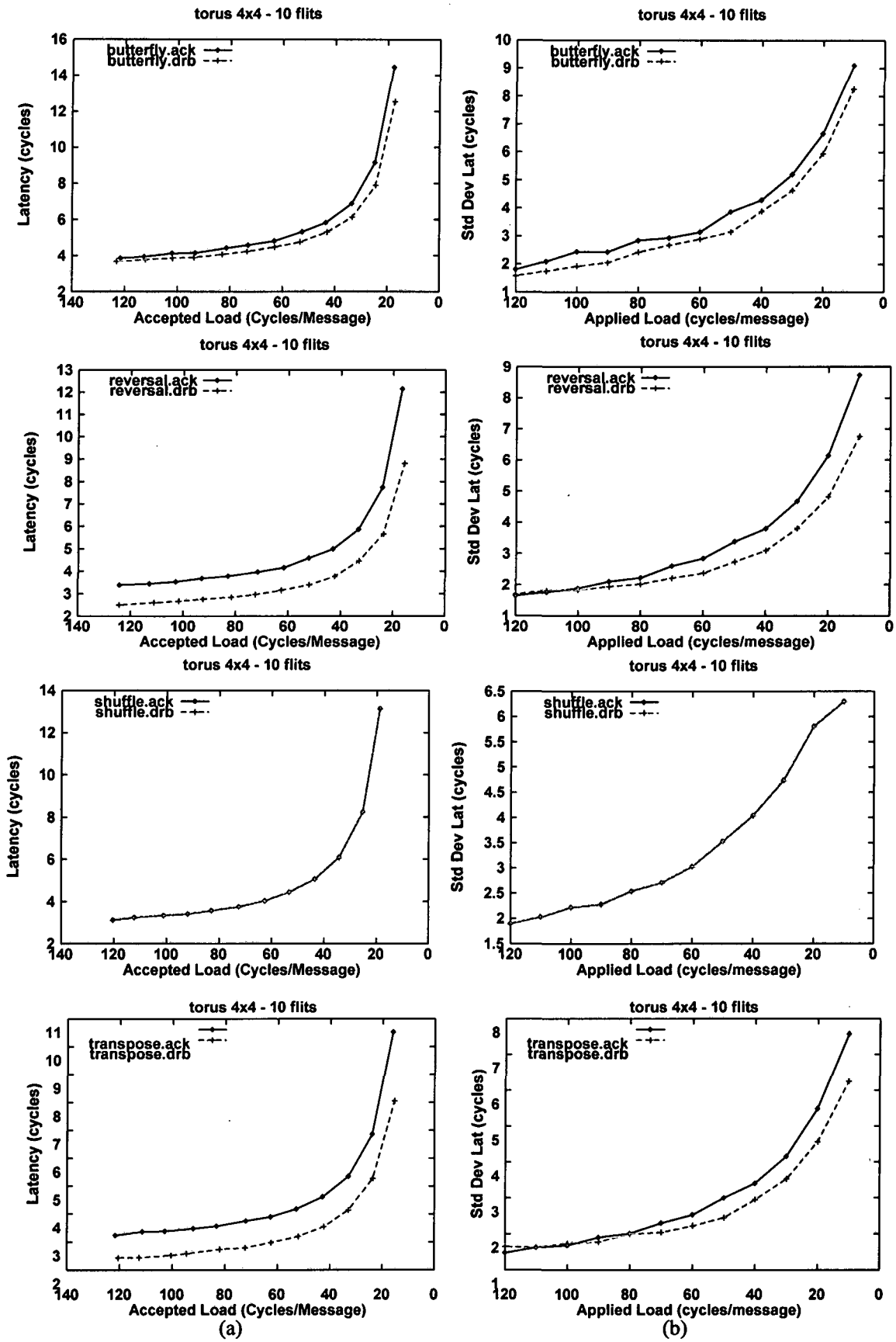


Figura 6-4 Rendimiento incluyendo el mensaje de reconocimiento en el Toro 4x4

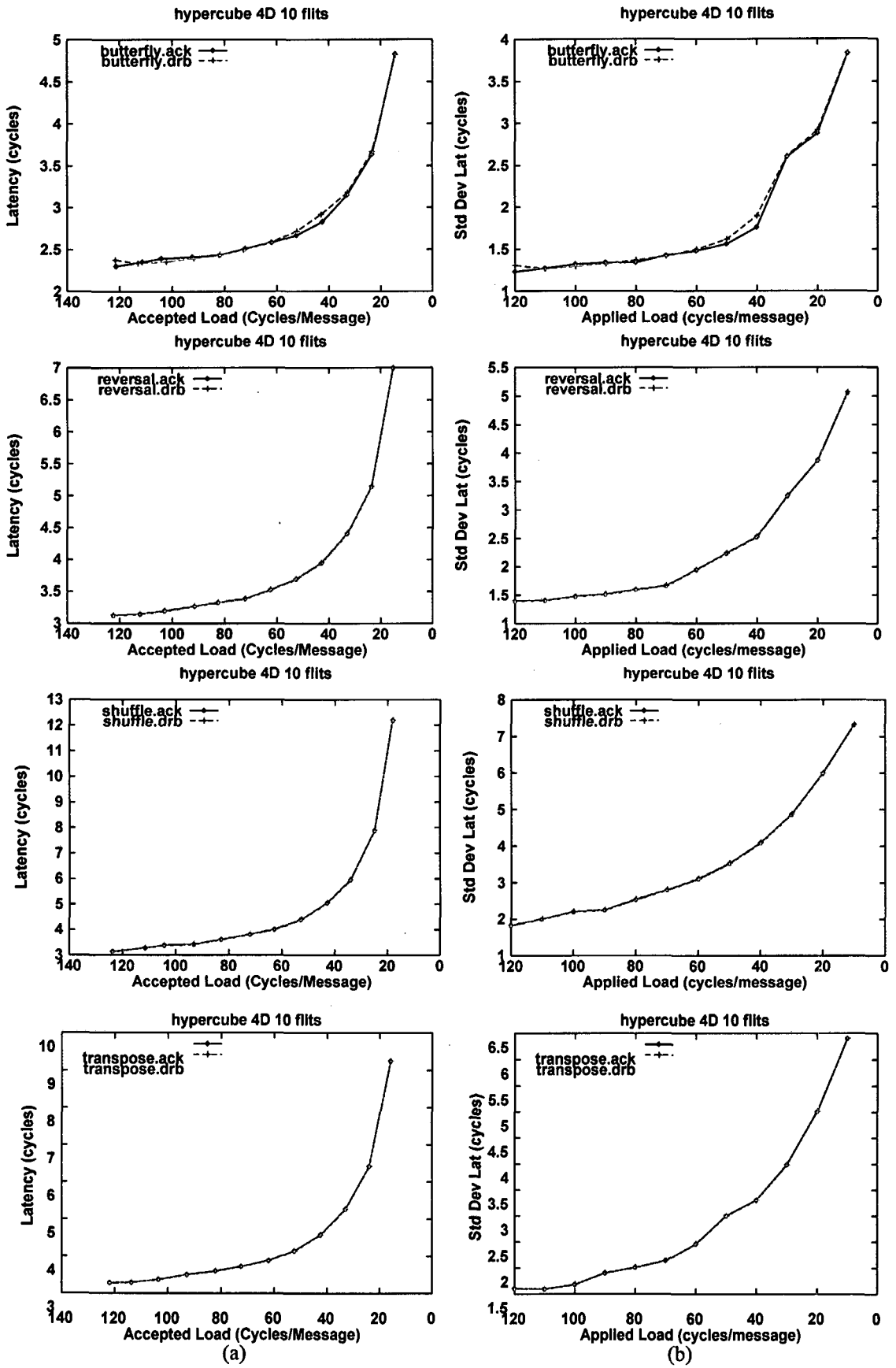
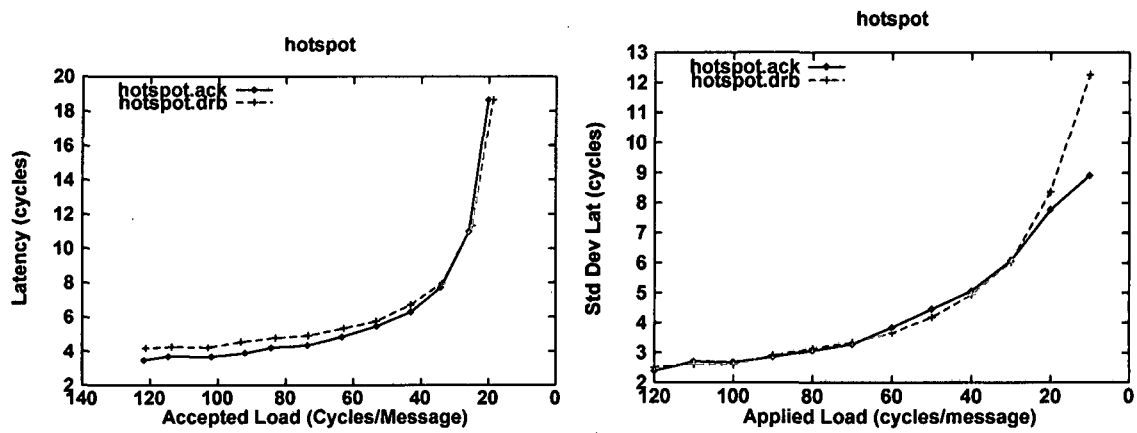


Figura 6-5 Rendimiento incluyendo el mensaje de reconocimiento en el Hipercubo 4D



(a) (b)

Figura 6-6 Rendimiento incluyendo el mensaje de reconocimiento para el patrón de "*hot-spot*"

6.4 Generación temprana del mensaje de reconocimiento

Es evidente que la pieza clave del funcionamiento adecuado de DRB es la información de monitorización que obtienen los mensajes y que se envía a los nodos origen. Como se ha comentado en el capítulo 4, el tiempo de respuesta de DRB y por lo tanto su capacidad para adaptarse a las condiciones de tráfico presentes en la red de interconexión en todo momento, depende de la vigencia de esa información de monitorización. Esta información se utiliza para conocer el estado actual de la red y predecir el estado en momentos inmediatamente posteriores. Suponiendo que la red de interconexión se comporta como un sistema continuo, se espera que las predicciones sean capaces de seguir el comportamiento futuro de la red en todo momento. Por todo ello, es muy importante que el tiempo de respuesta de DRB sea pequeño.

En este punto, queremos analizar la respuesta transitoria de DRB frente a la carga de comunicaciones. Para ello mostramos, para el patrón de "*hot-spot*" utilizado hasta ahora, la gráfica temporal de latencias sufridas en la red. Además, queremos estudiar la sensibilidad de DRB frente a dos aspectos de comportamiento.

Los dos aspectos a evaluar son las dos alternativas respecto el informe de la latencia del algoritmo básico de DRB presentadas en el capítulo 4. La primera es informar de la latencia cuando el mensaje de usuario llega al destino final. La segunda es la posibilidad de generar el mensaje de reconocimiento en cuanto la latencia acumulada de un camino multipaso supere un cierto intervalo en lugar de esperar a llegar al nodo destino. Como se vio en aquel punto, con esta política se pretende informar al nodo origen lo mas pronto posible de los problemas que puede haber sobre el metacamino que está utilizando. Recordemos que esta información anticipada debe

utilizarse solamente para configurar los supernodos y no para seleccionar los caminos multipaso, para los cuales se debe seguir utilizando el registro de latencias totales de los caminos multipaso. A esta alternativa la hemos llamado *generación temprana del mensaje de reconocimiento.*

Hemos realizado una serie de experimentos presentados aquí para tener un estudio exhaustivo de la sensibilidad de DRB a estos parámetros tanto de la latencia media conseguida como de la respuesta transitoria sufrida. La experimentación con la generación del mensaje de reconocimiento en el nodo destino la llamamos DRB y la experimentación con la generación temprana del mensaje de reconocimiento DRB2. La Tabla 6-2 muestra la experimentación realizada para mostrar la influencia de la generación temprana del mensaje de reconocimiento. El simulador funcional presentado en el capítulo 3 utilizado para esta experimentación incluye ambas posibilidades de manera seleccionable por el usuario.

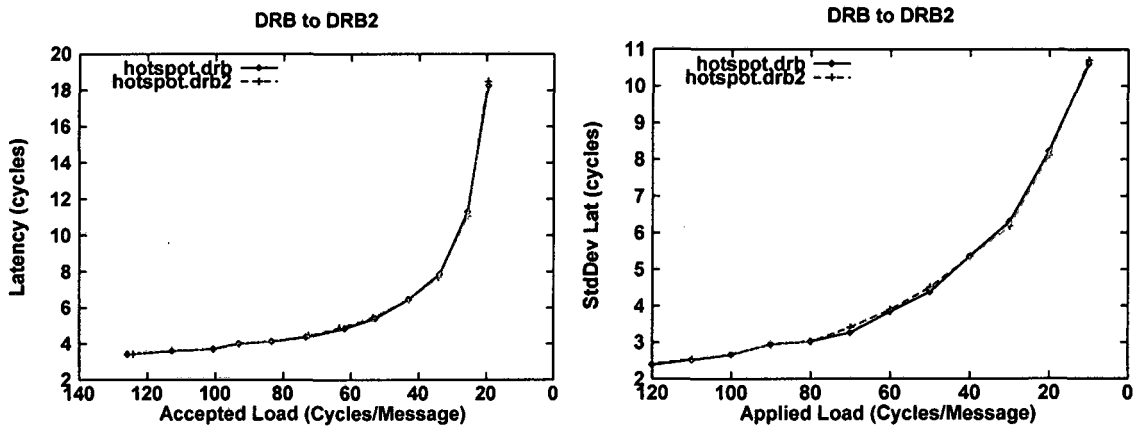
Topología	**Tamaño**	**Patrones**	**Encaminamiento**	**Análisis**	**Figura**
Toro	8x8 (64)	Hot-spot	DRB	Estacionario	Figura 6-7
			DRB2	Transitorio	Figura 6-8

Tabla 6-2 Experimentación para mostrar la influencia de la generación temprana del mensaje de reconocimiento

La Figura 6-7 muestra los valores de latencia media (a) y de desviación estándar de la latencia media (b) para cada uno de estos casos para el patrón "*hot-spot*" cuando la carga de comunicaciones varía desde 120 hasta 10 ciclos por mensaje. Como puede observarse, las curvas están prácticamente superpuestas, lo que demuestra poca o nula influencia de estos aspectos sobre los resultados medios en régimen estable cuando se utiliza DRB.

Asimismo, nos hemos fijado en la respuesta transitoria en estos experimentos. La Figura 6-8 muestra la respuesta temporal de la latencia media (a), la desviación estándar de la latencia (b) y el "*throughput*" conseguido (c) en los primeros 6000 ciclos de simulación, para una carga de comunicaciones de 10 ciclos por mensaje, que es el punto de carga máxima de la gráfica anterior. Se puede observar que los casos de generación temprana del mensaje de reconocimiento presentan una respuesta un poco mejor antes de los primeros 2000 ciclos de simulación eliminándose los valores máximos de latencia, pero que los resultados pronto convergen sobre un valor único.

Esto demuestra que con un número de mensajes suficientemente alto, el sistema converge y que, el tiempo de convergencia es similar para ambos, lo cual implica que no mejora el tiempo de convergencia al utilizar la generación temprana del mensaje de reconocimiento.



(a) (b)

Figura 6-7 Efecto de la generación temprana del mensaje de reconocimiento.

Es conveniente señalar además, que para tan poco tiempo de simulación, los resultados pueden tener poca precisión estadística ya que se tienen pocos valores a promediar. Por todo ello, podemos concluir que estos aspectos tampoco tienen gran influencia en los resultados de DRB sobre las comunicaciones en la red de interconexión.

6.5 Retraso del mensaje de reconocimiento

En este punto, queremos analizar la influencia del retraso del mensaje de reconocimiento en la red. Para ello mostramos, para el patrón de "*hot-spot*" utilizado hasta ahora, la gráfica temporal de latencias sufridas en la red. Queremos estudiar la sensibilidad de DRB frente al retardo del mensaje de reconocimiento que informa al nodo origen de la latencia sufrida por el camino multipaso.

Así pues, queremos analizar qué sucede cuando ese mensaje de reconocimiento se retarda un cierto tiempo viajando por la red antes de llegar al nodo origen. Se pueden adoptar dos soluciones para enviar los mensajes de reconocimiento por la red de interconexión que pueden generar este retraso:

- Los mensajes de reconocimiento viajan en la red normal de datos pero con una prioridad baja para no entorpecer a los mensajes de datos

- El ancho de banda específico dedicado a los mensajes de reconocimiento es pequeño con objeto de perturbar poco la parte de la red dedicada a los mensajes de datos

En cualquiera de las dos posibilidades, los módulos de configuración de caminos y de selección de caminos multipaso no tienen información tan actualizada del estado de la red y podemos decir que realizan decisiones con información vieja sobre una situación nueva. A este aspecto le llamamos *retardo del mensaje de reconocimiento.*

Hemos realizado una serie de experimentos presentados aquí para tener un estudio exhaustivo de la sensibilidad de DRB a estos parámetros tanto de la latencia media conseguida como de la respuesta transitoria obtenida. El simulador funcional presentado en el capítulo 3 utilizado para esta experimentación incorpora la posibilidad de forzar un retardo al mensaje de reconocimiento de manera seleccionable por el usuario. Sobre el retardo del mensaje de reconocimiento, hemos seleccionado una serie de retardos: retardo 0 ciclos (w0), retardo 5 ciclos (w5), retardo 10 ciclos (w10), retardo 15 ciclos (w15) y retardo 20 ciclos (w20). La Tabla 6-3 muestra la experimentación realizada para mostrar la influencia del retraso del mensaje de reconocimiento.

Topología	**Tamaño**	**Patrones**	**Retardo**	**Análisis**	**Figura**
Toro	8x8 (64)	Hot-spot	0 ciclos 5 ciclos	Estacionario	Figura 6-9
			10 ciclos 15 ciclos 20 ciclos	Transitorio	Figura 6-10

Tabla 6-3 Experimentación para mostrar la influencia del retraso del mensaje de reconocimiento

La Figura 6-9 muestra los valores de latencia media (a) y de desviación estándar de la latencia media (b) para cada uno de estos casos para el patrón "*hot-spot*" cuando la carga de comunicaciones varía desde 120 hasta 10 ciclos por mensaje. Como puede observarse, todas las curvas están de nuevo prácticamente superpuestas, lo que

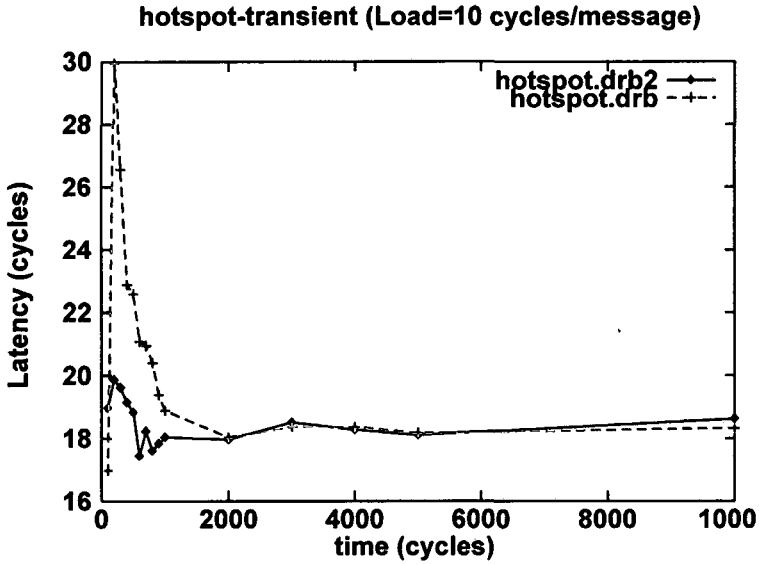
demuestra nula influencia de estos aspectos sobre los resultados medios en régimen estable cuando se utiliza DRB.

Asimismo, nos hemos fijado en la respuesta transitoria en estos experimentos. La Figura 6-10 muestra la respuesta temporal de la latencia media (a), la desviación estándar de la latencia (b) y el "*throughput*" conseguido (c) en los primeros 6000 ciclos de simulación, para una carga de comunicaciones de 10 ciclos por mensaje, que es el punto de carga máxima de la gráfica anterior. Se puede observar que los casos de generación temprana del mensaje de reconocimiento sin retardo o con un retardo pequeño presentan una respuesta ligeramente mejor antes de los primeros 2000 ciclos de simulación, pero que los resultados pronto convergen sobre un valor único. En este caso tampoco se modifica el tiempo de convergencia, ya que sobre los 2000 ciclos de simulación (Figura 6-10) todos los valores de latencia son muy parecidos.

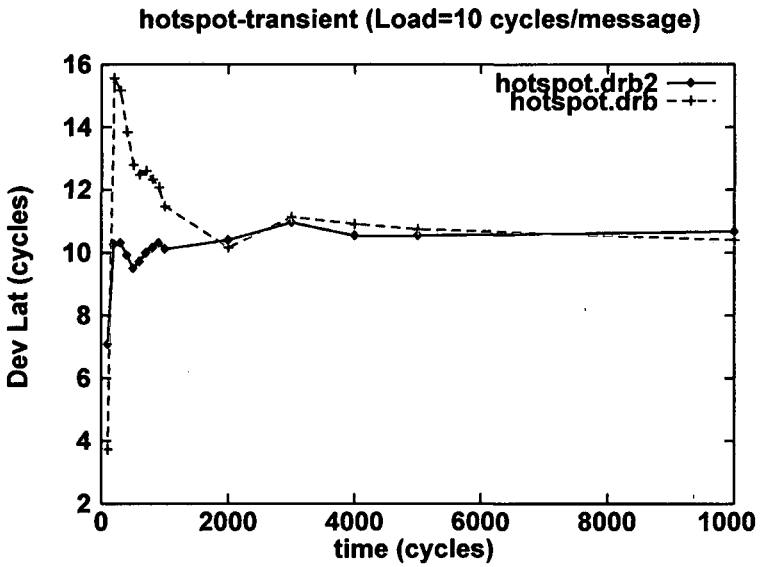
Es conveniente señalar además, que para tan poco tiempo de simulación, el número de mensajes que circula por la red es muy pequeño, en cuyo caso los resultados pueden tener poca precisión estadística y pueden aparecer sesgados ya que se tienen pocos valores a promediar. Por todo ello, podemos concluir que este aspecto tampoco tiene gran influencia en los resultados de DRB sobre las comunicaciones en la red de interconexión, lo cual muestra la robustez de DRB ante la información que precisa para trabajar.

Como conclusión de los tres experimentos 6-3, 6-4 y 6-5, podemos comentar que, teniendo en cuenta los resultados de los experimentos 6-4 y 6-5 que muestran la nula influencia de la generación temprana del mensaje de reconocimiento ni del retraso del mismo, que la mínima influencia mostrada en la Figura 6-4 para la topología toro desaparecería, pues no colisionarían los mensajes de ACK con los de datos, con lo cual se puede trabajar como si los mensajes de ACK no existieran. Para ello, se debería adoptar una de las dos soluciones apuntadas en el punto 6-5 respecto a los mensajes de reconocimiento y se debe tener en cuenta asimismo que, aunque se haga una reserva de un cierto porcentaje del ancho de banda de la red para los mensajes de reconocimiento, esta reserva se vería compensada por el incremento del "*throughput*" que permite DRB.

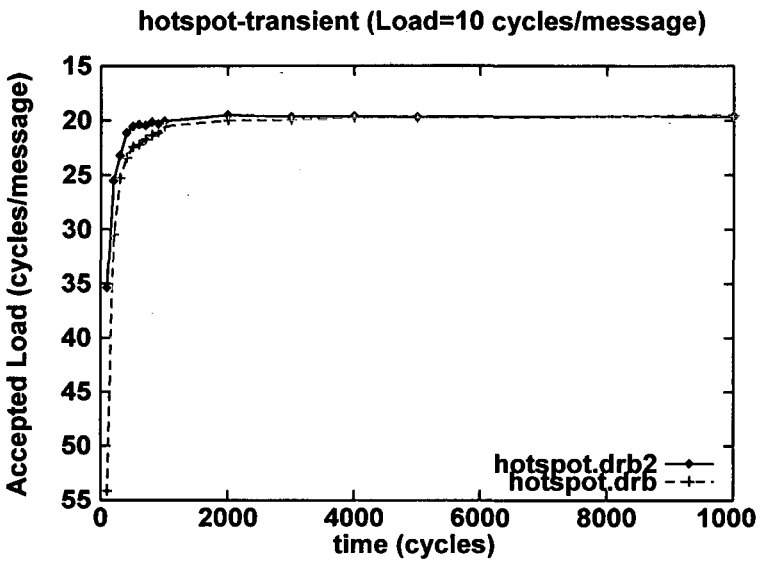
Esta conclusión es muy importante, pues pone de manifiesto la ausencia de penalización por monitorizar (DRB es un sistema basado en la monitorización del estado de la red) o bien, que el hecho de introducir una monitorización genera una ganancia suficiente para justificar el gasto introducido.



(a)

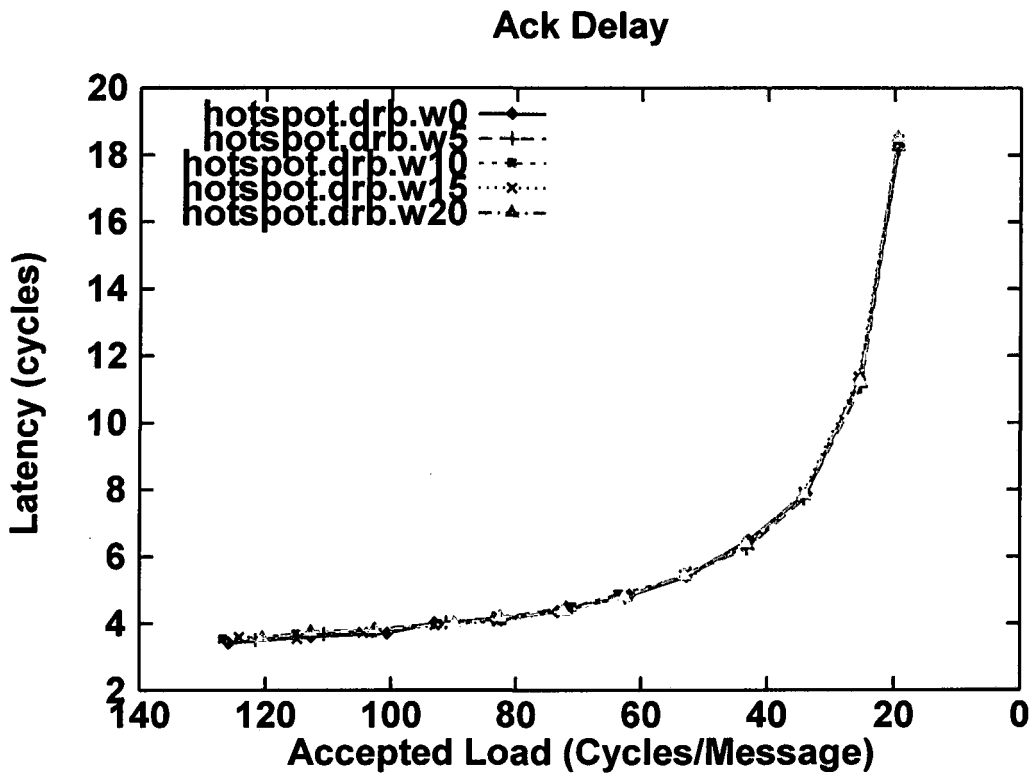


(b)

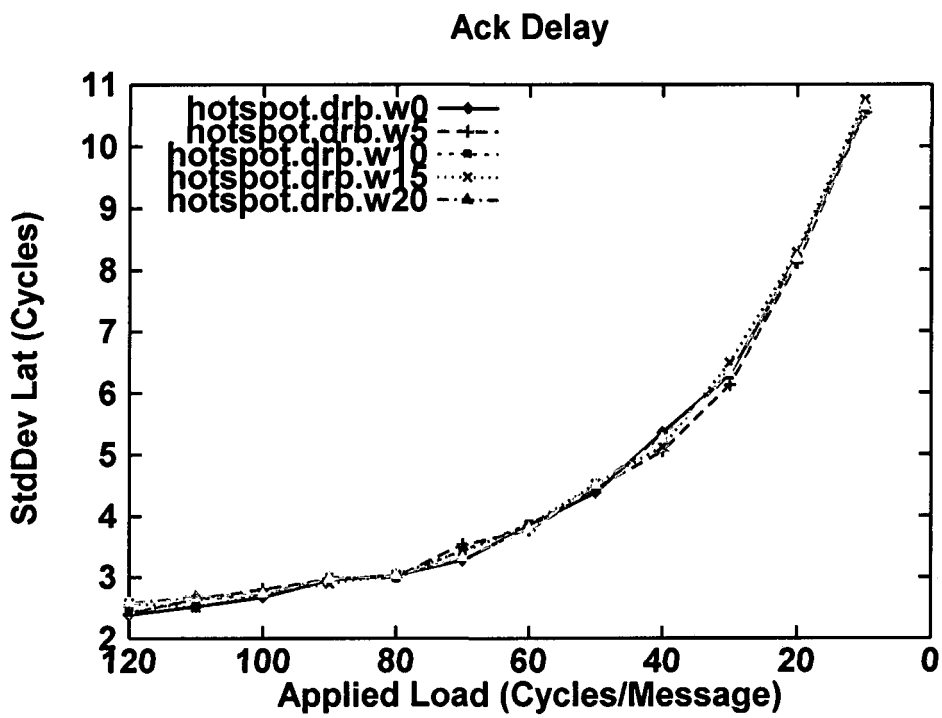


(c)

Figura 6-8 Análisis transitorio del efecto de la generación temprana del mensaje de reconocimiento

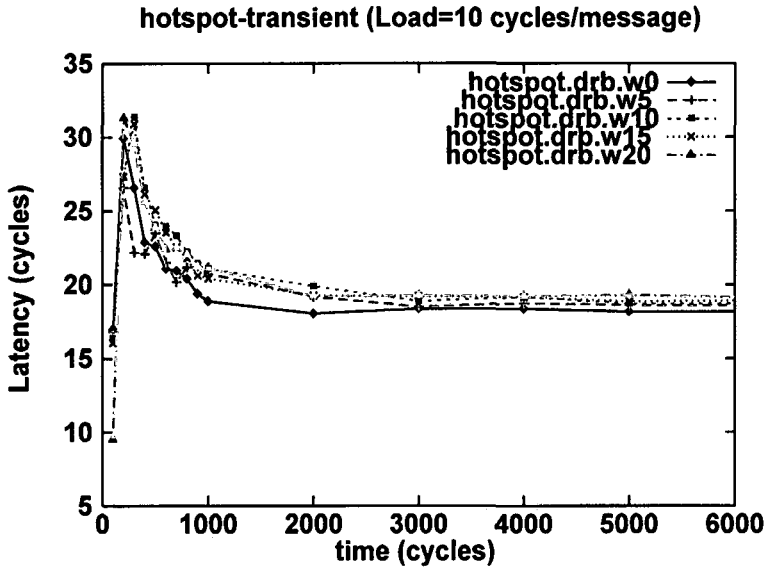


(a)

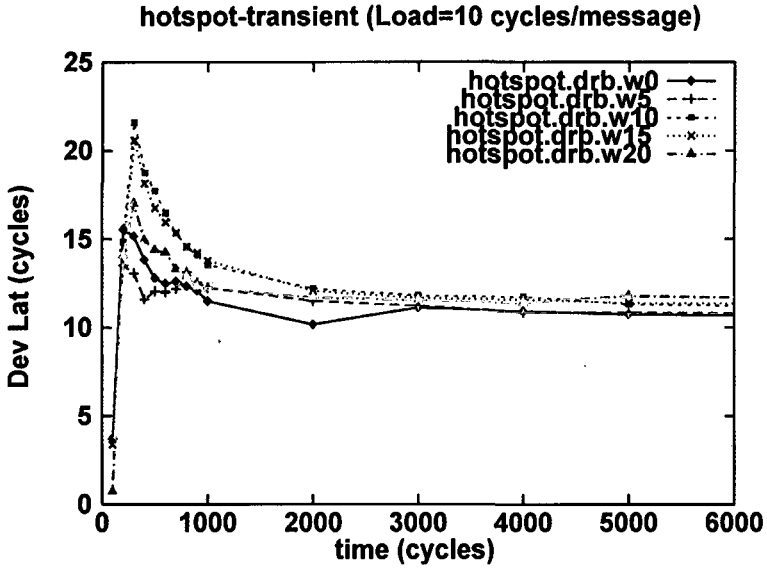


(b)

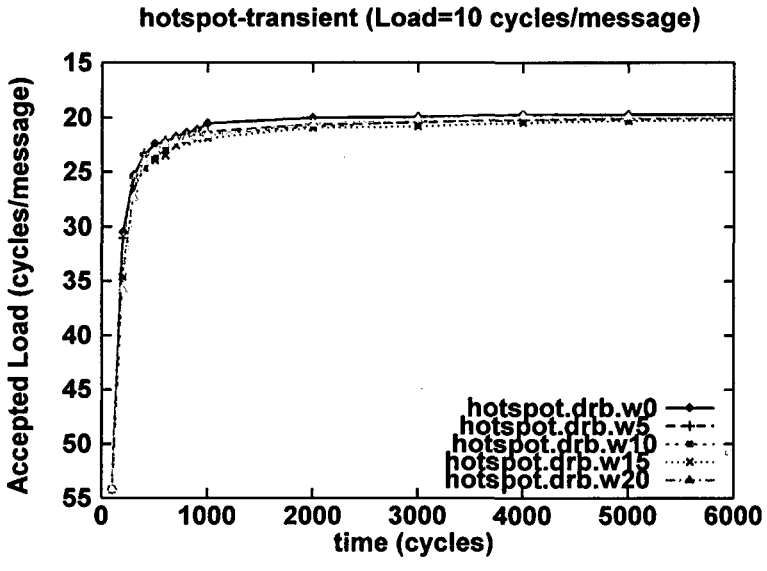
Figura 6-9 Efecto del retardo del mensaje de reconocimiento.



a)



b)



c)

Figura 6-10 Análisis transitorio del efecto de retardo del mensaje de reconocimiento.

6.6 Experimentación con patrones de comunicación

En este apartado se presentan los resultados de la experimentación realizada sobre patrones de comunicación, y que ha sido dividida en dos apartados: los patrones sistemáticos, por un lado, y los patrones específicos, por el otro. Esta experimentación se ha realizado sin tener en cuenta la influencia del mensaje de reconocimiento, generando el mismo al llegar al nodo destino final y sin retardar su comunicación al nodo fuente. Esta es una configuración que, dado que los experimentos anteriores han mostrado la poca influencia del mensaje de reconocimiento en la red de interconexión, nos permite simular con el caso más sencillo sin que los resultados se aparten de los valores "reales".

6.6.1 Experimentación con patrones sistemáticos

Los experimentos se han realizado para toros e hipercubos para dos tamaños: 16 y 64 nodos. Para los cuatro patrones de comunicación, el desbalanceo de la carga de mensajes de comunicación es grande y los comportamientos del encaminamiento estático, adaptativo completo y DRB difieren en gran manera. La Tabla 6-4 muestra la experimentación realizada con los patrones sistemáticos de comunicación.

<table>
<tr><th>Topología</th><th>Tamaño (Nodos)</th><th>Patrones</th><th>Encaminamiento</th><th>Figura</th></tr>
<tr><td rowspan="2">Toro</td><td>4x4 (16)</td><td rowspan="4">Butterfly

Reversal

Shuffle

Transpose</td><td rowspan="4">Estático

Adaptivo

DRB</td><td>Figura 6-11</td></tr>
<tr><td>8x8 (64)</td><td>Figura 6-12</td></tr>
<tr><td rowspan="2">Hipercubo</td><td>4D (16)</td><td>Figura 6-13</td></tr>
<tr><td>6D (64)</td><td>Figura 6-14</td></tr>
</table>

Tabla 6-4 Experimentación con patrones sistemáticos de comunicación

6.6.1.1 Topología Toro

La Figura 6-11 muestra los resultados de latencia media (a) y desviación estándar (b) de la red respecto la carga aceptada de mensajes para los patrones *"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"* cuando se utilizan los encaminamientos estático (sta), adaptativo (ada) y DRB (drb) sobre un toro de 16 nodos 2D. Los resultados son similares para los cuatro patrones, así que los comentaremos conjuntamente.

En general, DRB ofrece mejores resultados que la aproximación adaptativa y, por supuesto, la determinista, que nos sirve como umbral mínimo. El caso determinista representa la cota superior, ya que es estático y no utilizan información para adaptarse. La diferencia entre las curvas para cada patrón se incrementa a medida que se incrementa la carga de tráfico de la red. Se puede observar que a "baja carga" (un intervalo entre mensajes mayor que 80 ciclos), DRB se comporta de manera similar a los otros encaminamientos. Esto significa que DRB no cambia el comportamiento de la red cuando no es necesario, de manera que no introduce ningún "overhead" extra.

Cuando la red trabaja a tasas de "carga intermedia", entre 80 y 40 ciclos entre mensajes, con DRB el incremento de la latencia (Figura 6-11 (a)) es notablemente inferior al que sufre con los otros métodos porque empieza a usar caminos alternativos. A "cargas altas" de la red, para intervalos menores que 40 ciclos entre mensajes, DRB usa el mayor número de caminos multipaso permitidos en la configuración, resultando en valores de latencia menores respecto a los otros métodos. Para todos los casos de latencia mostrados, su desviación estándar (Figura 6-11 (b)) es menor en el caso de DRB que en los demás métodos de encaminamiento, lo cual muestra una mayor uniformización de las latencias, que es uno de los objetivos de DRB.

Además, a medida que la carga se incrementa, las reducciones proporcionales de latencia también se incrementan mostrando la mayor capacidad de DRB frente a los otros métodos. En las tasas de carga mayor, se consiguen reducciones de latencia del orden del 50 por ciento o mayor respecto de los otros métodos. Al mismo tiempo que estas latencias se reducen, el "*throughput*" conseguido se mejora. Es decir, el número de mensajes entregados con menor latencia se incrementa. Este aspecto se observa en las gráficas de latencia presentadas (Figuras (a)), en las que en el eje x se presenta la carga aceptada, que es una función de la carga aplicada. Se observa que cada punto de DRB representa una carga mayor que el correspondiente punto de los otros métodos de encaminamiento porque se sitúa más a la derecha en el gráfico. Los otros métodos se comportan peor porque, con ellos, la red se satura antes y, por tanto, consiguen valores menores de carga aceptada.

Para mostrar la validez del método de encaminamiento DRB, las redes consideradas se calcularon también para un tamaño mayor de 64 nodos. La Figura 6-12 muestra los resultados para las redes toroidales bidimensionales de 64 nodos. En el caso del toro 8x8 la respuesta de DRB frente a los otros métodos de encaminamiento es todavía mejor que para el toro 4x4. Se observa que el retroceso en las gráficas cuando se aplica una carga alta es grande para los casos estático y adaptivo, lo que significa que existe rechazo de carga, mientras que en DRB la carga aceptada se mantiene constante

(*"Butterfly"* y *"Perfect Shuffle"*) o se incrementa ligeramente (*"Bit-Reversal"* y *"Matrix Transpose"*). En el caso del *Shuffle* para el adaptivo, el resultado es incluso peor que el estático porque este patrón, para esta topología toro 8x8, no deja caminos mínimos libres y el hecho de usar adaptatividad empeora la situación.

Hemos encontrado resultados similares para redes de 16 y 64 nodos que mejoran al aumentar la red de tamaño, lo cual significa una buena escalabilidad de DRB frente a los otros métodos. Además, en algunos casos se observa que las gráficas sufren un retroceso sobre la carga aceptada, lo que significa que la red se satura totalmente a partir de un cierto punto a partir del cual el incremento de la carga lo único que provoca es el incremento del número de mensajes rechazados. Esta saturación se alcanza para cargas mayores en DRB lo cual permite su utilización en un rango mayor de carga antes de aparecer el fenómeno de la saturación.

Como conclusión general para el caso del toro, se puede observar que DRB se comporta mejor que los otros métodos de encaminamiento, sobre todo cuando la carga es muy alta, en cuyo caso la carga rechazada debido a la saturación de la red es mayor en otros métodos que en DRB. DRB presenta latencias del orden de la mitad en el caso del toro 4x4 (*"Butterfly"*, *"Bit-Reversal"* y *"Matrix Transpose"*) hasta 4 veces menores (caso del patrón *"Bit-Reversal"* en el toro 8x8). Por tanto, en conjunto, DRB presenta una menor latencia y un mayor rango de utilización (se satura a cargas más altas), por un lado, y tiene valores de desviación estándar de la latencia menores, con lo cual la uniformidad de la latencia es mayor, por el otro. Estos dos resultados son los objetivos perseguidos por DRB.

6.6.1.2 Topología Hipercubo

El mismo tipo de experimentación se ha realizado para la topología de hipercubo. Se han encontrado resultados similares, en los cuales DRB ofrece mejores resultados que los métodos estático y adaptativo para los hipercubos de diferentes tamaños. La Figura 6-13 y Figura 6-14 muestran los resultados para los hipercubos de 4 y 6 dimensiones, respectivamente.

En el caso del hipercubo 4D (Figura 6-13) el comportamiento de DRB y adaptativo son muy similares, pero DRB es mejor en la zona de carga máxima, en la que presenta menores latencias. En el caso de *"Butterfly"* ambos métodos se comportan de manera similar. Para el caso de *"Bit-Reversal"*, son muy similares menos en el último punto de máxima carga en la que DRB ofrece menor latencia. Esto significa que DRB es capaz de soportar cargas mayores. Resultados similares se obtienen para *"Perfect Shuffle"*, en el que DRB mejora a adaptativo en los dos últimos puntos de carga. En el

caso de *"Matrix Transpose"*, por la forma como el patrón de comunicaciones se acopla sobre la topología y el número de caminos mínimos alternativos (en este caso, todos los caminos alternativos son mínimos), DRB ofrece resultados ligeramente peores que el caso adaptativo, menos en el último punto en el que sí que lo supera. Nuevamente, demuestra que, ante condiciones extremas, DRB ofrece mejores prestaciones que otros métodos debido a la distribución de caminos utilizada. En el caso de las desviaciones estándar de estas latencias (Figura 6-13 (b)), DRB presenta valores similares a los otros métodos de manera generalizada, aunque ligeramente favorables para DRB excepto en el caso del patrón "*Butterfly*".

Como en el caso del toro, al aumentar el tamaño de la red y considerar el hipercubo de 6D (Figura 6-14), los resultados son cualitativamente similares al hipercubo de 4D, pero las diferencias son todavía mas acusadas. En el caso *"Butterfly"*, adaptivo y DRB presentan comportamientos similares, algo mejores para adaptivo en cargas bajas y medias. Esto es debido a la combinación del tipo de patrón y topología. En el caso de *"Bit-Reversal"* y *"Matrix Transpose"*, se comportan muy similares hasta que en los dos últimos puntos la latencia con el caso adaptivo se dispara mientras que con DRB se incrementa ligeramente. Esto es debido a que, a cargas altas, los caminos mínimos se saturan y el método adaptivo no es capaz de reducir la latencia. En el caso *"Perfect Shuffle"*, sucede como el caso de 4D, DRB es mejor que adaptivo, el cual es incluso peor que el estático. Como se ha comentado, esto es debido a que la asignación de este patrón sobre el hipercubo, hace que los posibles caminos mínimos de un canal estén ocupados por otros canales y el hecho de introducir adaptatividad, lo que hace es incrementar el número de colisiones y empeorar la situación en la red. Las desviaciones estándar de las latencias (Figura 6-14 (b)) se comportan de manera similar, excepto a alta carga que el caso del adaptivo se disparan a valores altos mientras que en DRB se mantienen uniformes. Como puede observarse, sobre la topología hipercubo, DRB presenta resultados aceptables. Sobre esta topología, por sus características físicas de distancia media y grado, las latencias que se producen son mucho menores que en el caso del toro, con lo que DRB no puede mejorar significativamente los resultados. Sobre la topología toroidal, que tiene la característica de ser fácilmente escalable, DRB consigue valores de latencia mucho menores que el caso adaptativo. Se puede observar que DRB consigue ganancias mucho mayores respecto al adaptivo sobre el toro que sobre el hipercubo.

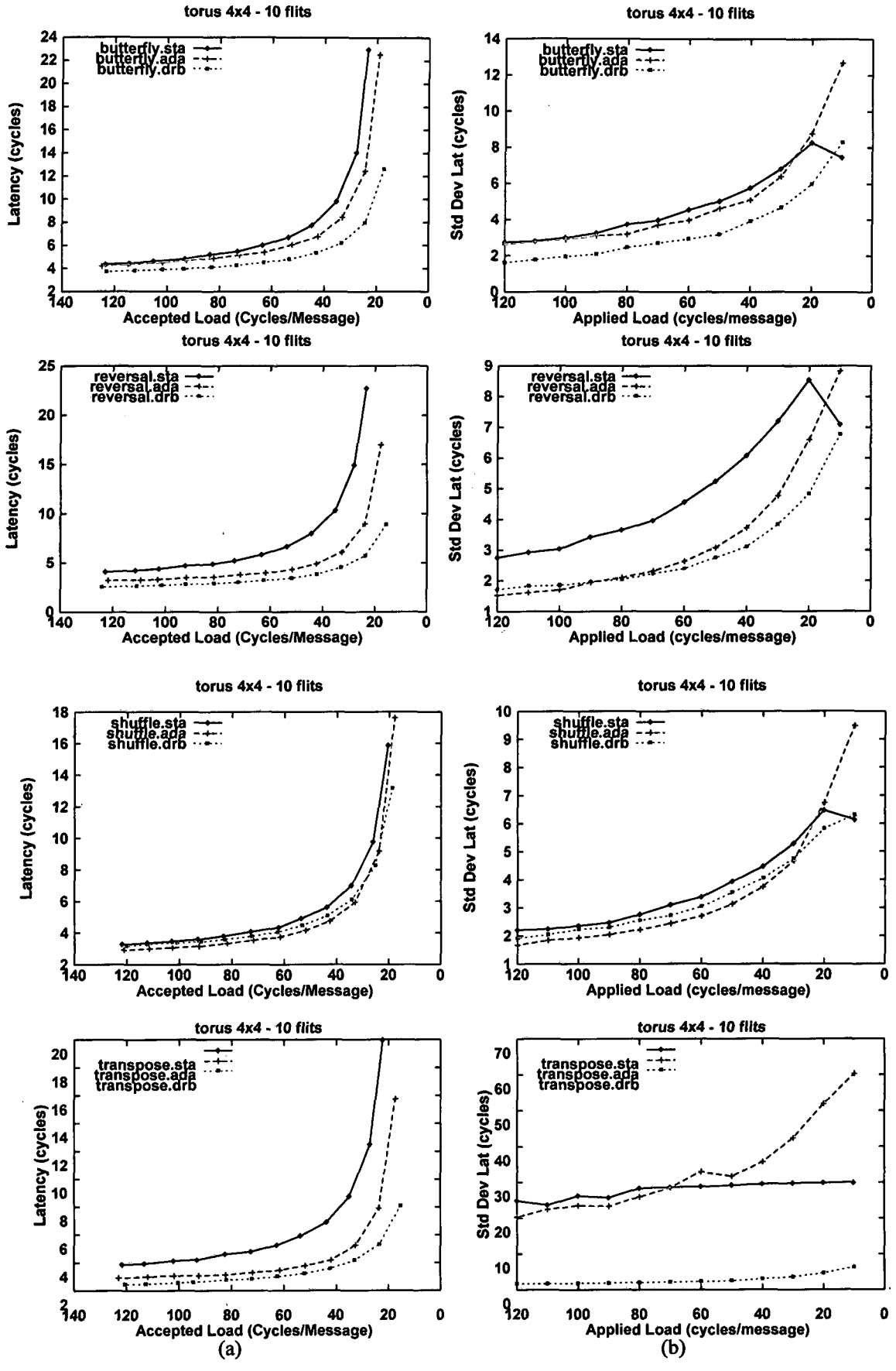


Figura 6-11 Rendimiento para los diferentes patrones en el Toro 4x4

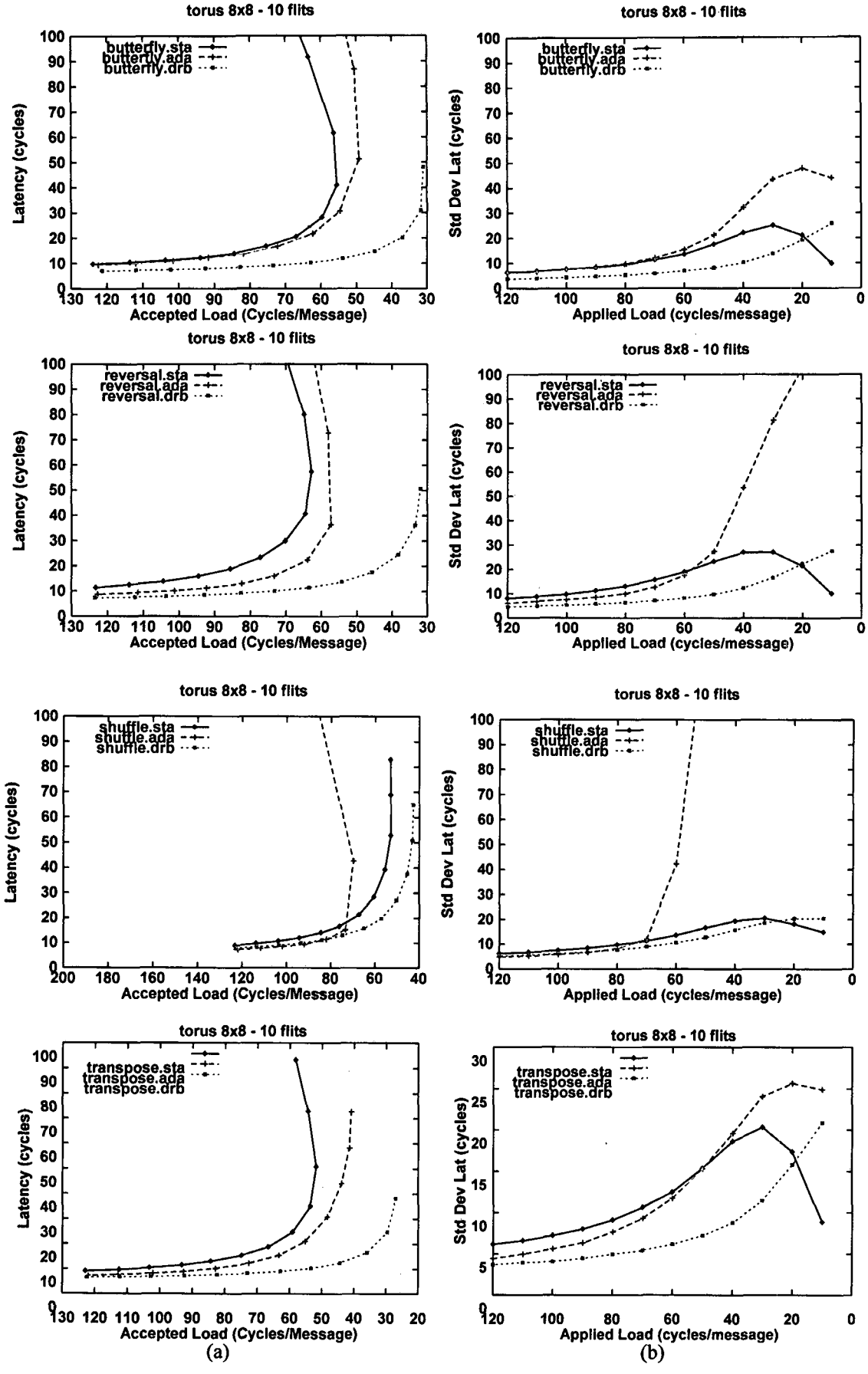


Figura 6-12. Rendimiento para los diferentes patrones en el Toro 8x8

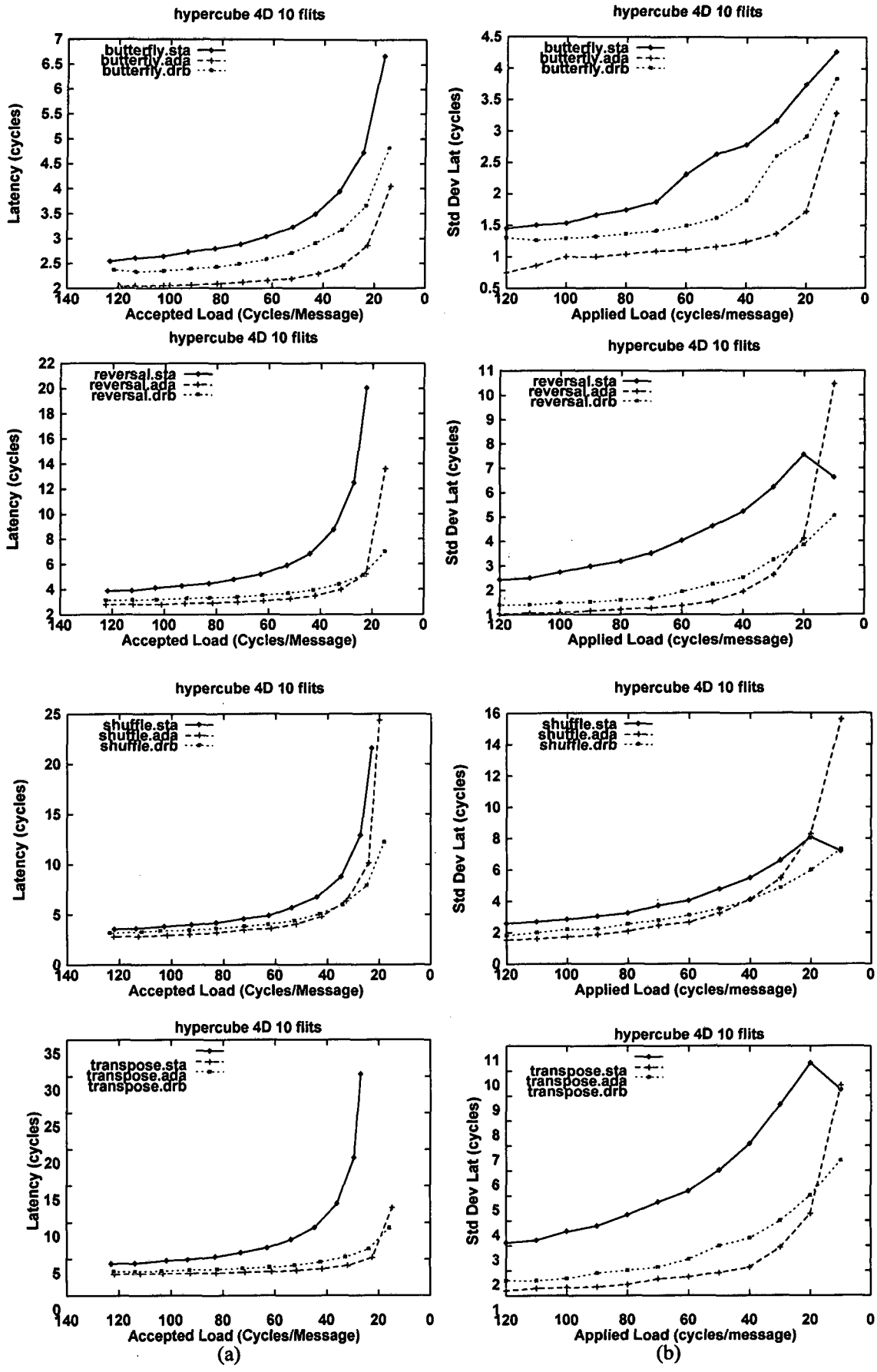


Figura 6-13 Rendimiento para los diferentes patrones en el Hipercubo 4D

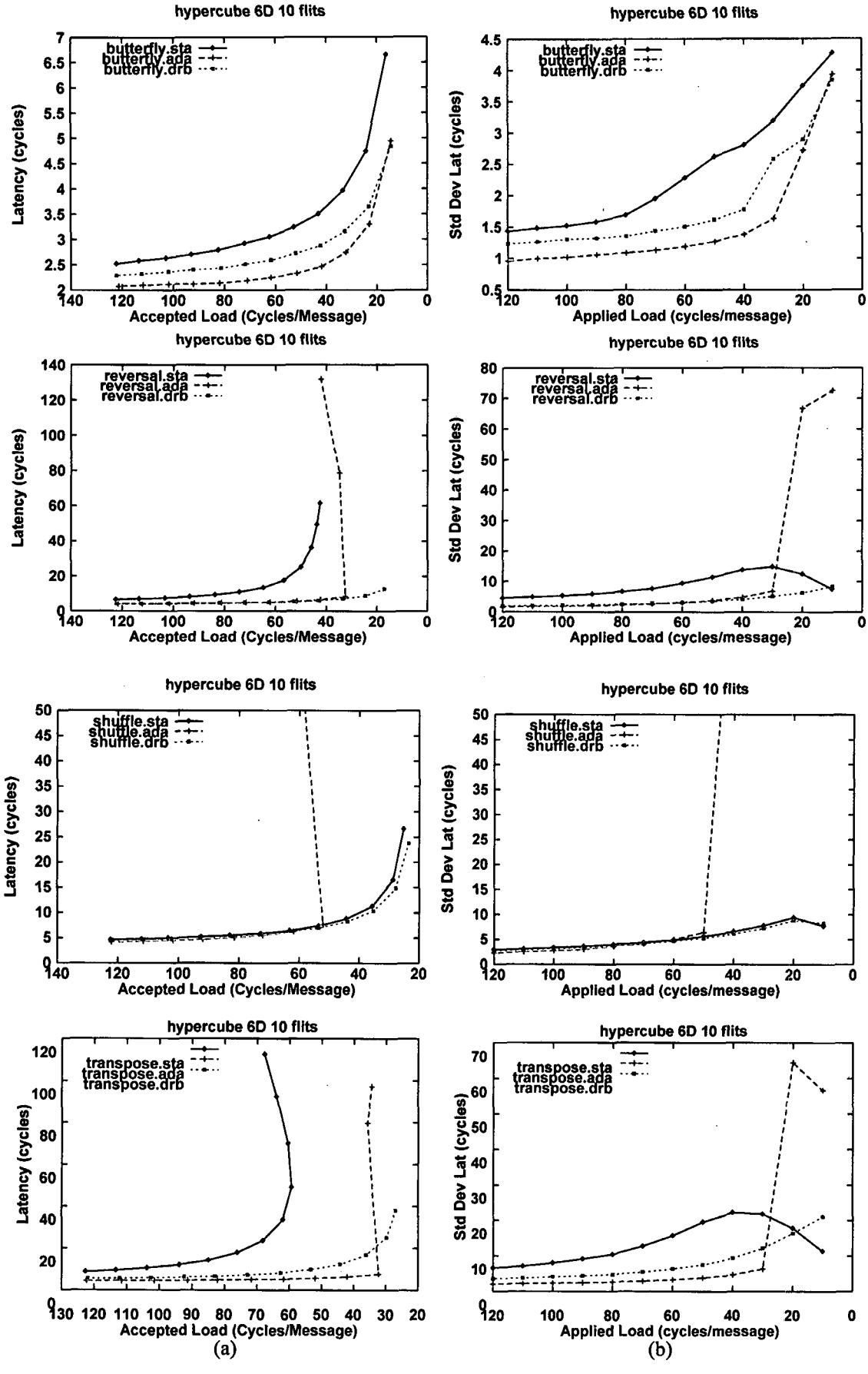


Figura 6-14 Latencia para los diferentes patrones en el Hipercubo 6D

6.6.2 Experimentación con patrones específicos: "*hot-spot*"

Con objeto de analizar y comparar DRB frente a los otros dos algoritmos de encaminamiento en condiciones extremas, hemos diseñado un experimento donde el patrón de comunicación definido provoca la aparición de "*hot-spot*", en el que varios mensajes colisionan sobre un camino común como se ha comentado anteriormente en este capítulo. El patrón de "*hot-spot*" presentado en la Figura 6-1 supone condiciones de comunicación muy rigurosas. La Tabla 6-5 muestra la experimentación realizada sobre el patrón específico "hot-spot".

<table>
<tr><th>Topología</th><th>Tamaño</th><th>Patrones</th><th>Encaminamiento</th><th>Análisis</th><th>Figura</th></tr>
<tr><td rowspan="4">Toro</td><td rowspan="4">8x8 (64)</td><td rowspan="4">Hot-spot</td><td>Estático
DRB
Adaptivo</td><td>Estacionario</td><td>Figura 6-15</td></tr>
<tr><td>Estático</td><td rowspan="3">Mapa de latencia</td><td>Figura 6-16</td></tr>
<tr><td>DRB</td><td>Figura 6-17</td></tr>
<tr><td>Adaptivo</td><td>Figura 6-18</td></tr>
</table>

Tabla 6-5 Experimentación sobre el patrón específico "hot-spot"

La Figura 6-15 muestra los resultados de latencia (a) y desviación estándar de la latencia (b) para cada uno de los tres algoritmos de encaminamiento. Como se puede ver en la figura, DRB mejora los resultados frente a los otros algoritmos ya que consigue menores latencias (hasta una tercera parte menores en la zona de carga máxima aplicada (10 ciclos/mensaje), la latencia de DRB es cercana a 20 ciclos mientras que la del adaptivo de 60 ciclos) mientras se incrementa el "*throughput*" (el doble que el caso adaptivo cuando la carga aplicada es de 10 ciclos por mensaje, DRB acepta mensajes hasta una tasa de 20 ciclos/mensaje mientras que el adaptivo acepta solo mensajes cada 40 ciclos) y, por lo tanto, se consigue una tasa mayor de tráfico aceptado en la red. La latencia también es más uniforme porque las desviaciones estándar son menores en el caso de DRB (Figura 6-15 (b)). La disminución de las desviaciones de las latencias en los casos estático y adaptivo en el punto de carga máxima (10 ciclos/mensaje) es debida

a la gran saturación que existe en la red que provoca que todos los mensajes sufran una latencia muy alta similar.

La Tabla 6-6 muestra los resultados para el experimento con el patrón "*hot-spot*" para el punto de carga máxima de las gráficas anteriores (10 ciclos/mensaje), donde se observa la gran reducción en latencia (de mas de una tercera parte) e incremento en "*throughput*" (del orden del 100% frente al adaptivo) de DRB frente a los otros métodos. Estos resultados son debidos a las condiciones de tráfico que presenta el patrón "*Hot-spot*", en las cuales los caminos mínimos están ocupados por mensajes de datos pertenecientes al propio patrón y el método adaptivo no es capaz de reducir tanto la latencia como el método DRB que utiliza caminos no mínimos poco cargados.

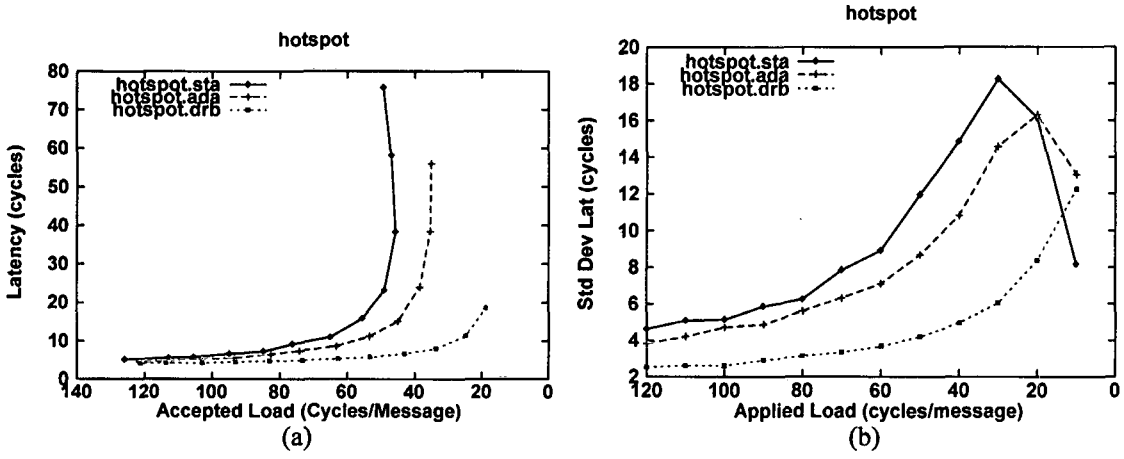


Figura 6-15 Rendimiento para el patrón de "*hot-spot*"

Hot-spot (Carga aplicada = 10 ciclos/mensaje)					
Método de Encaminamiento	**Latencia Media (ciclos)**	**Desviación Estándar (ciclos)**	**Latencia Máxima (ciclos)**	***Throughput* (%)**	**C.Apli./C.Ace. (ciclos/men.)**
Estático	75.84	8.17	86.0	24%	10/50
Adaptivo	55.86	13.3	86.25	40%	10/35
DRB	18.64	12.25	116.75	57%	10/20

Tabla 6-6 Resultados para el experimento con el patrón "*hot-spot*"

Con objeto de mostrar el efecto colectivo de DRB en la distribución de la carga y la desaparición de los "*hot-spots*", en la Figura 6-16, Figura 6-17 y Figura 6-18

mostramos las gráficas de "superficie de latencia" para los enlaces de la red de interconexión para el patrón de "*hot-spot*" que estamos estudiando. En estas figuras, se muestra la latencia cuando la carga de paquetes está establecida en un intervalo entre paquetes de 10 ciclos de simulación, que es el punto de carga máxima de la Figura 6-15. Cada punto de la rejilla muestra la latencia media de los enlaces de un nodo del toro utilizado en el patrón "*hot-spot*".

Se puede observar que, cuando se usa encaminamiento estático (Figura 6-16), aparecen grandes valores máximos de latencia mientras otras zonas de la red están siendo escasamente utilizadas. La latencia media de los paquetes que han viajado por la red en este caso es de 75.84 y la relación entre la carga aceptada y la carga aplicada es del 24%.

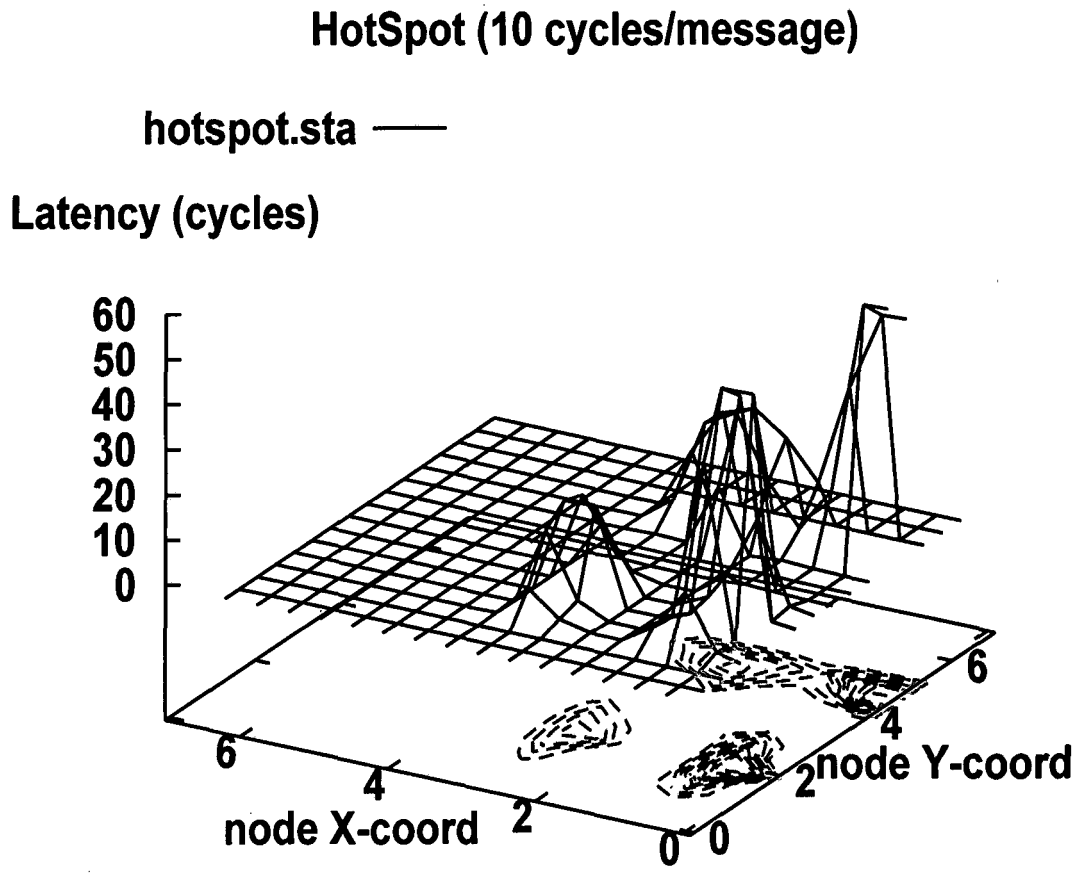


Figura 6-16 Distribución de la latencia en la red para el patrón de comunicaciones que provoca la aparición del "*hot-spot*" utilizando encaminamiento estático

El encaminamiento adaptativo (Figura 6-17) reduce la latencia media hasta 55.86 ciclos y mejora el porcentaje de carga aceptada hasta un 40%, pero no es capaz de eliminar totalmente los "*hot-spots*".

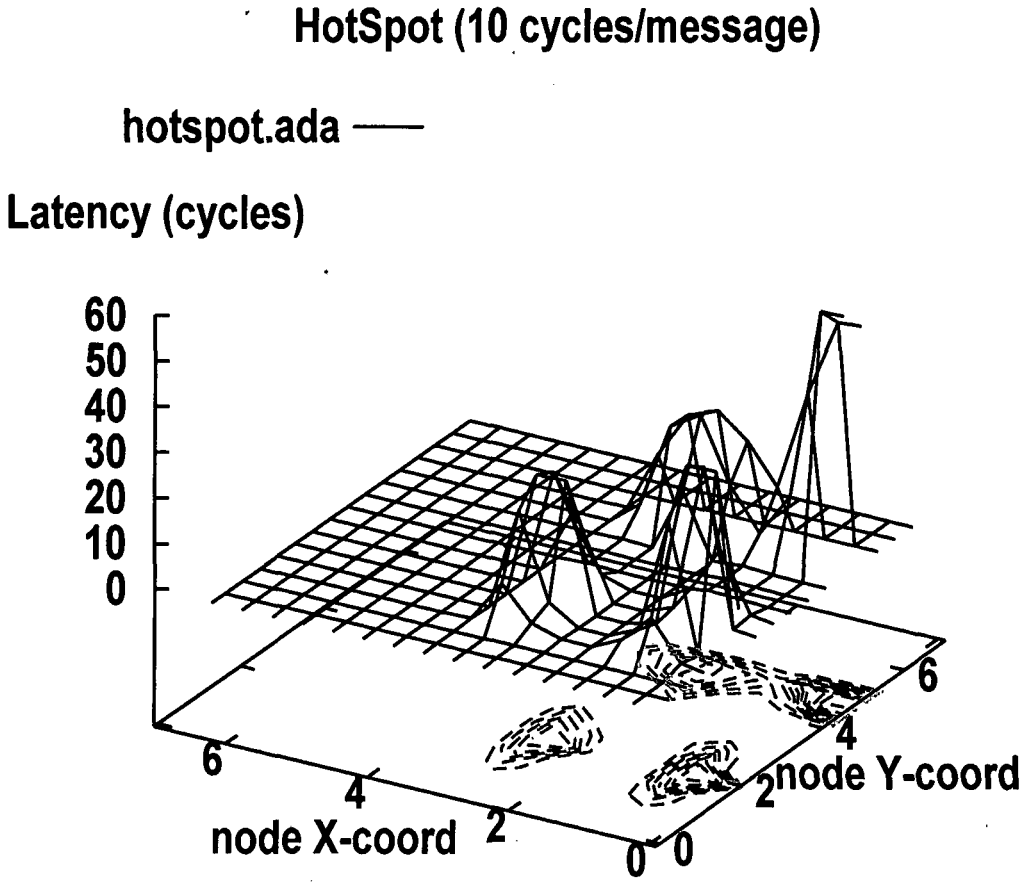


Figura 6-17 Distribución de la latencia en la red para el patrón de comunicaciones que provoca la aparición del "*hot-spot*" utilizando encaminamiento adaptivo

Cuando se usa el método DRB (Figura 6-18), se eliminan efectivamente estos picos de latencia ("*hot-spots*") porque el exceso de carga en ellos se distribuye sobre

otros enlaces. La latencia media en este caso es de 18.64 ciclos y la carga aceptada del 57%. La distribución efectiva de la carga de comunicaciones conseguida por DRB frente a los otros métodos se puede observar sobre las isolíneas de contorno proyectadas sobre la base de cada una de las figuras. Esta proyección muestra la mayor utilización de los enlaces de la red de interconexión conseguida por DRB y el consiguiente menor incremento de las latencias en la red. Los mapas para el caso estático y adaptivo son similares. En el caso del adaptivo, aunque utiliza más enlaces que el estático, no se consiguen valores de latencia tan reducidos como en DRB debido a que los caminos mínimos están saturados.

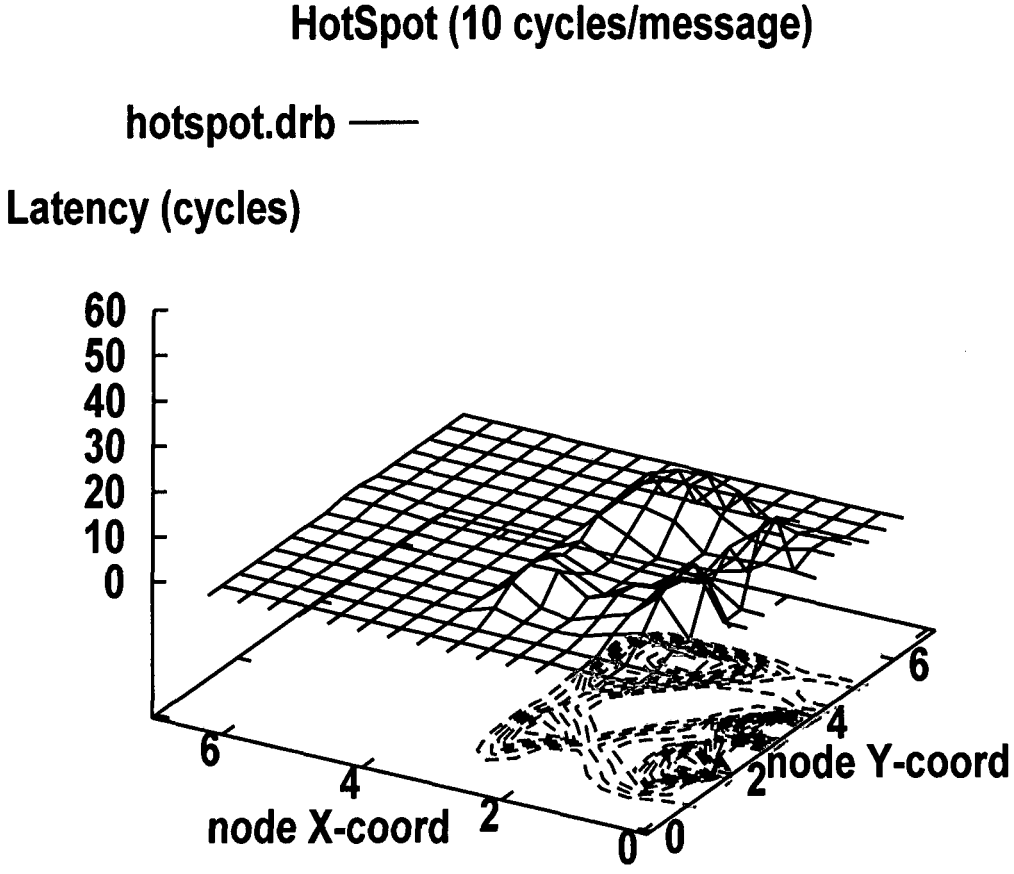


Figura 6-18 Distribución de la latencia en la red para el patrón de comunicaciones que provoca la aparición del "*hot-spot*" utilizando encaminamiento DRB

La conclusión de estos datos es que DRB ofrece mejores resultados para patrones tipo "*hot-spot*" de gran concentración local de carga ya que es capaz de distribuir el exceso de carga de unos nodos a otros nodos menos cargados y balancear de manera efectiva la carga de comunicaciones en toda la red de interconexión.

6.7 Influencia de la longitud del paquete

Cuando se trata con aplicaciones reales es importante conocer la influencia de la longitud del paquete que se envía sobre la latencia del sistema de comunicaciones. Para ello, hemos realizado una experimentación consistente en variar la longitud del paquete. Así, hemos tomado longitudes de 10, 20 y 64 *"flits"*. Hay que tener en cuenta que el incremento de longitud del paquete supone un aumento efectivo de la carga de comunicaciones en la red. La Tabla 6-7 muestra la experimentación realizada para estudiar la influencia de la longitud del paquete.

Topología	**Tamaño (Nodos)**	**Patrones**	**Longitudes**	**Figura**
Toro	4x4 (16)	Butterfly	10 *"flits"* 20 *"flits"* 64 *"flits"*	Figura 6-19
		Reversal		Figura 6-20
		Shuffle		Figura 6-21
		Transpose		Figura 6-22

Tabla 6-7 Experimentación de la influencia de la longitud del paquete

Se puede observar, en la Figura 6-19, que las mejoras de rendimiento que DRB presenta frente a los otros métodos se incrementan a medida que se incrementa el tamaño del paquete, para el patrón de *"Butterfly"*. Este comportamiento indica una buena respuesta de DRB ante diversas longitudes de paquete y es debido a que DRB es capaz de sacar partido del incremento de carga que supone el incremento de la longitud del paquete para este patrón de comunicaciones.

La Figura 6-20 muestra los resultados para el patrón de *"Bit-Reversal"* para las diferentes longitudes de paquete. En este caso, DRB no mejora a medida que se incrementa el tamaño, sino que llega a presentar resultados ligeramente peores que el adaptativo, aunque lejos del encaminamiento estático. Esto es debido a que DRB tiene menores oportunidades de distribuir los mensajes al aumentar el tamaño de éstos para este patrón, ya que pocos mensajes ocupan más tiempo los enlaces del nodo de inicio, dificultando la distribución. El encaminamiento adaptativo, al permitir giros en cada encaminador, presenta mayores posibilidades de usar caminos alternativos.

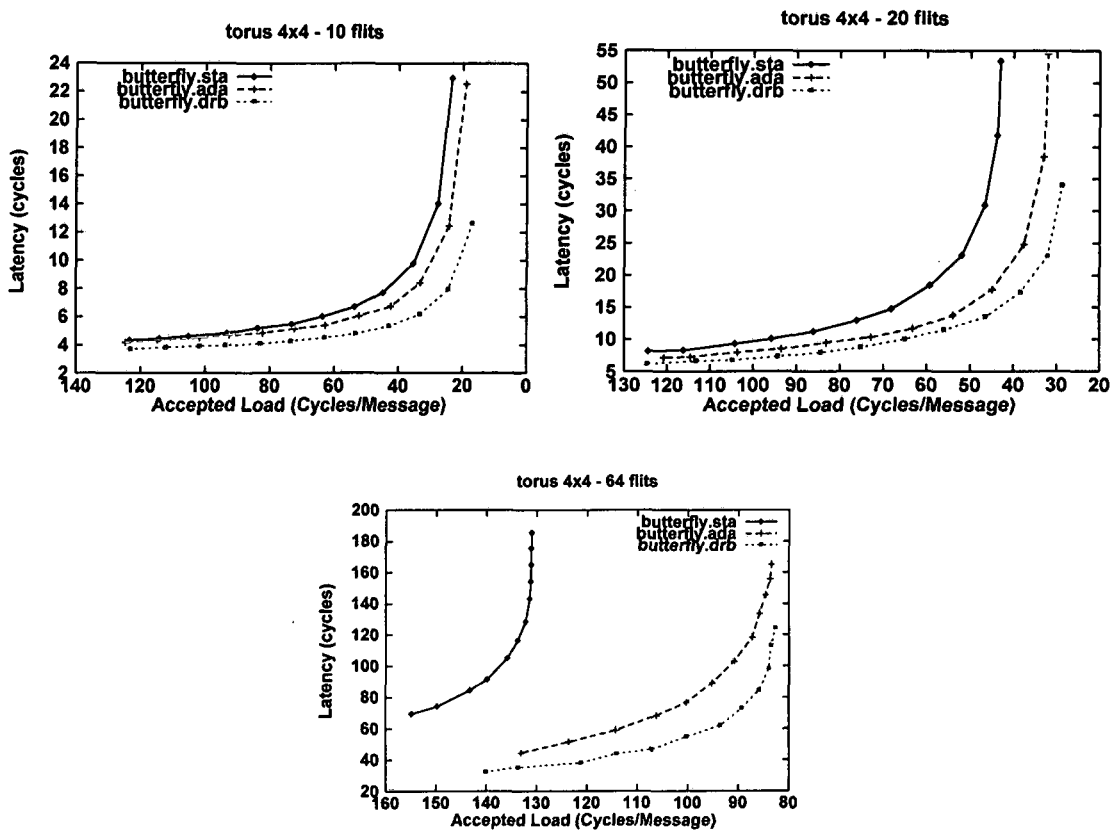


Figura 6-19. Rendimiento para diferentes longitudes del paquete para el patrón "Butterfly"

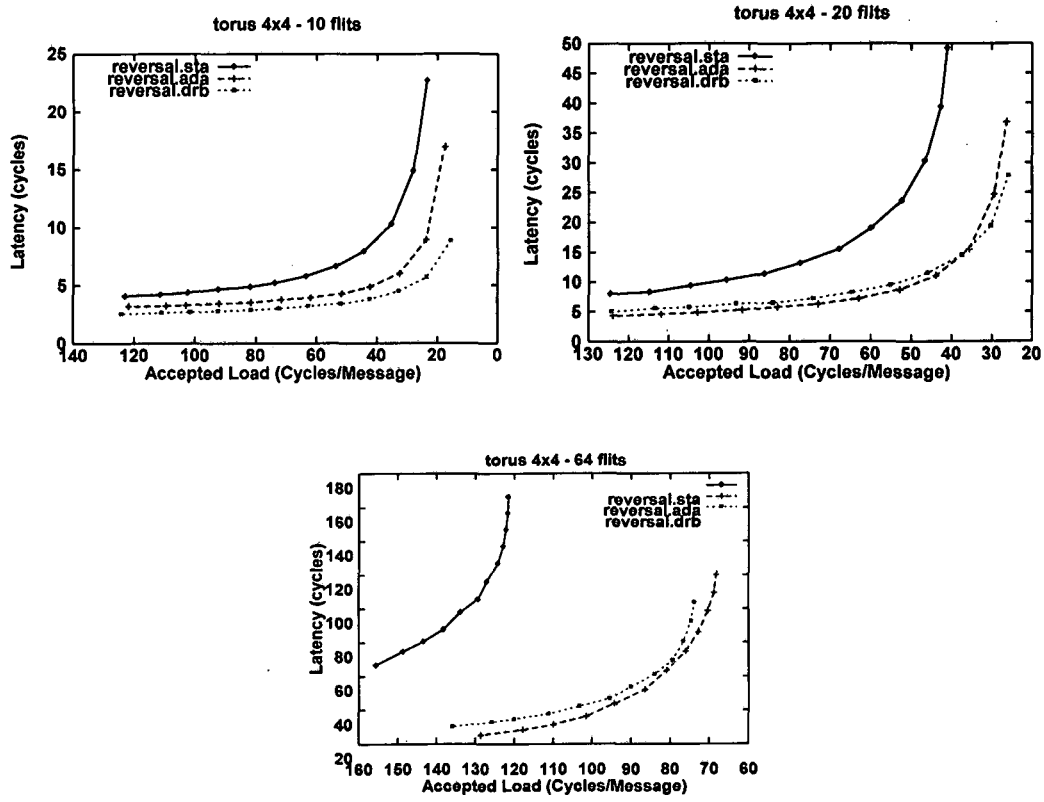


Figura 6-20. Rendimiento para diferentes longitudes del paquete para el patrón *"Bit-Reversal"*

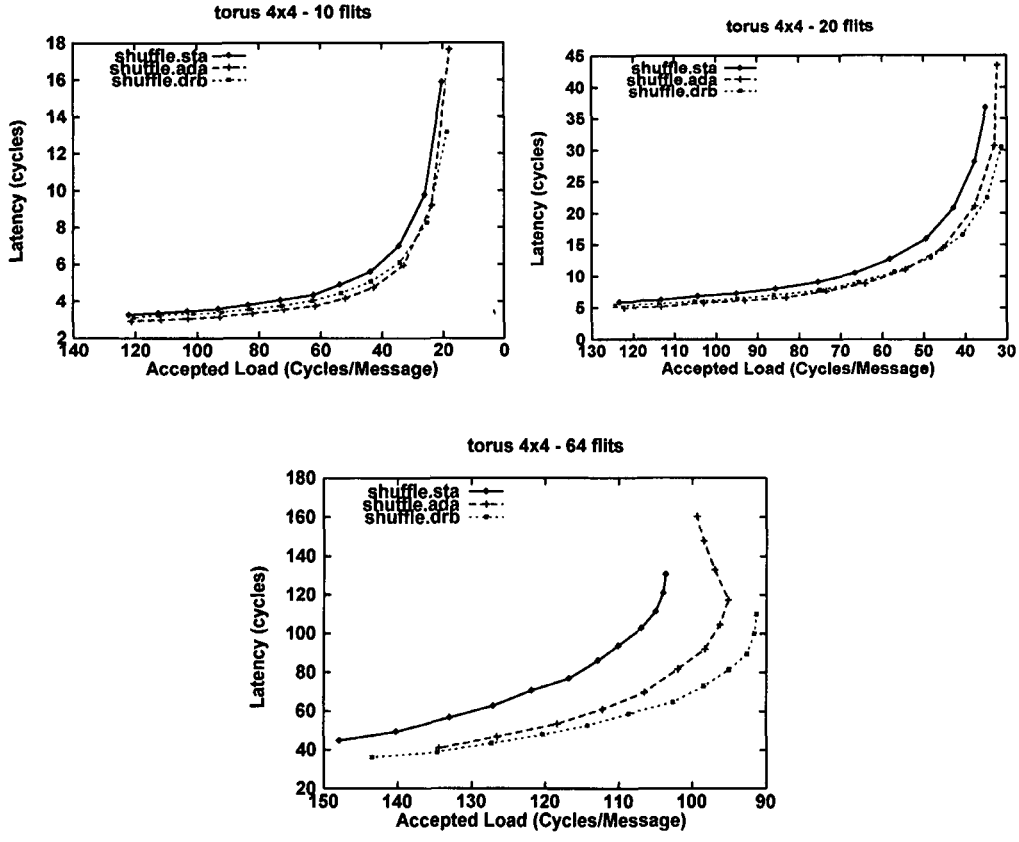


Figura 6-21. Rendimiento para diferentes longitudes del paquete para el patrón Per. Shuffle

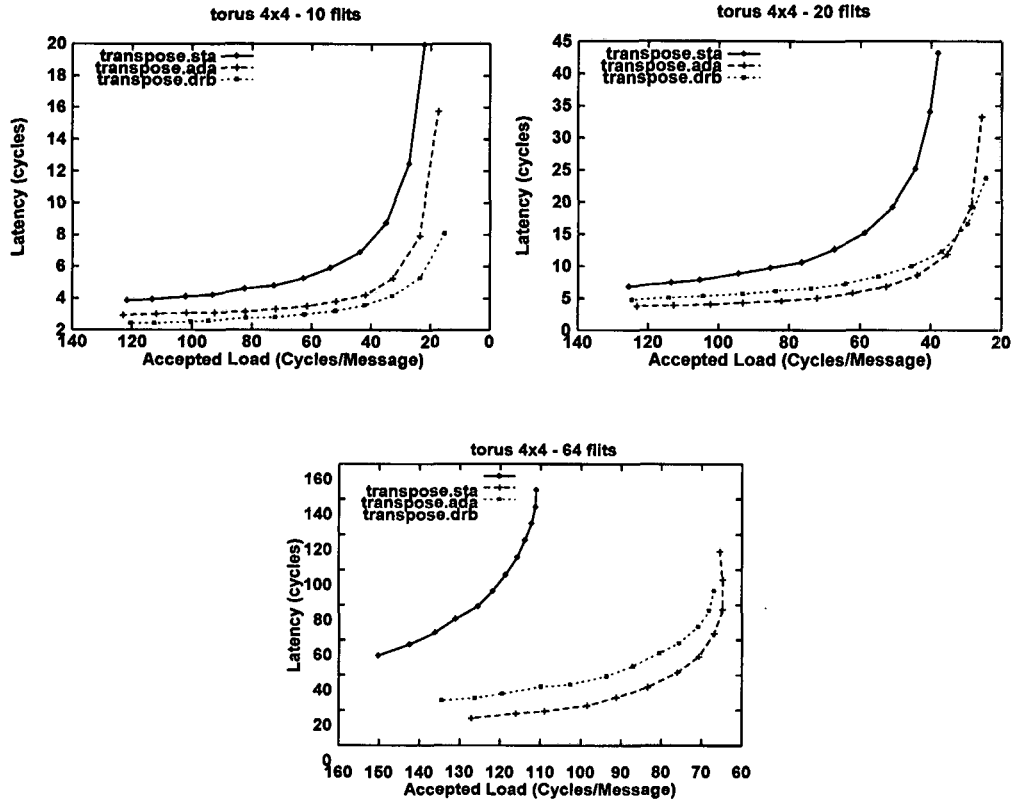


Figura 6-22. Rendimiento para diferentes longitudes del paquete para el patrón Mat. Transpose

La Figura 6-21 muestra los resultados de evaluar diferentes longitudes de paquete para el patrón de *"Perfect Shuffle"*. En este caso, DRB ofrece mejores resultados que los otros métodos en todos los casos. Únicamente, en el caso del tamaño más pequeño (10 *"flits"*), se mantienen los tres métodos muy similares y DRB sólo mejora en los casos de carga muy alta. También, en este caso, como en "*Butterfly*", DRB es capaz de aprovechar el incremento de longitud de los mensajes.

La Figura 6-22 muestra los resultados de diferentes longitudes de paquete para el patrón *"Matrix Transpose"*. En este caso, como para el patrón de *"Bit-Reversal"*, DRB no mejora los resultados frente al encaminamiento adaptativo, pero si que lo hace respecto al encaminamiento estático. Este comportamiento tiene su razón de ser en que, debido a la configuración del patrón sobre la topología, el incremento de la longitud del mensaje hace que se ocupen mucho tiempo los caminos desde el nodo fuente y no sea tan efectiva la distribución de mensajes realizada por DRB.

Como conclusión general, se puede observar que el hecho de aumentar la longitud del mensaje no es una buena opción, ya que hace aumentar la latencia para los tres métodos de encaminamiento de manera general. Para DRB, dependiendo del patrón, se mejora en algunos casos (*"Butterfly"* y *"Perfect Shuffle"*) y en otros no (*"Bit-Reversal"* y *"Matrix Transpose"*). Para el patrón "*Butterfly*", la influencia es pequeña, de manera que los resultados son similares para todos los tamaños de paquete. Para el patrón "*Bit-Reversal*", el aumento del tamaño del paquete no mejora los resultados de DRB, los cuales incluso son ligeramente peores que el caso adaptivo. Para el patrón "*Perfect Shuffle*", el incremento de la longitud del paquete hace obtener a DRB mejores resultados, es decir, mayores diferencias con el adaptivo. Para el patrón "*Matrix Transpose*", DRB ofrece peores resultados que el método adaptivo a medida que se incrementa la longitud del paquete

6.8 Escalabilidad de DRB frente a la topología y la aplicación

Hasta ahora hemos mostrado una evaluación de DRB bajo múltiples aspectos del sistema de comunicaciones: el patrón, la topología, el nivel de carga de la red, la longitud del mensaje. Asimismo, hemos evaluado el tiempo de respuesta de DRB analizando su comportamiento transitorio bajo una serie de alternativas como son retardar el envío del mensaje de reconocimiento o generar de manera temprana el mensaje de reconocimiento.

El objetivo de este punto es mostrar la escalabilidad de DRB cuando se escala la topología y/o la aplicación. Para ello, se muestran los resultados de evaluar un patrón concreto para una topología dada mientras se varía el tamaño y/o la dimensión de la

misma. Hemos elegido dos patrones de comunicación: un patrón representativo de los patrones sistemáticos como es el patrón de *"Butterfly"* y el otro patrón seleccionado es el patrón de "*hot-spot*". Nos interesa representar los dos casos en que el patrón escala con la topología y en el que no escala. En el caso que el patrón escale significa que, al aumentar la red, el patrón también aumenta y representa la misma carga que para la red de tamaño menor. En el caso de que el patrón no escala igual que la topología, al aumentar ésta, cambia la carga aplicada en la red.

Las topologías elegidas son, como en el resto de la experimentación, toros e hipercubos de la categoría de n-cubos k-arios. El patrón de "hot-spot", como se explica más adelante, escala con las dos topologías, mientras que el patrón "*Butterfly*" escala con el hipercubo, pero no con el toro, con lo que tenemos un muestrario de todos los casos posibles. La Tabla 6-8 muestra la experimentación realizada en este apartado. El caso de encaminamiento estático se muestra porque representa la opción de coste cero, es decir, usar la red de interconexión sin ningún sistema de mejora añadido. No se incluye el encaminamiento adaptativo al ser un análisis intrínseco de las capacidades de DRB. El caso estático se presenta como referencia. Del análisis de los resultados, veremos que, en algunos casos, el uso de DRB permite disminuir el grado de la red manteniendo el número de nodos, lo cual es una reducción de coste que puede compensar el coste de añadir la propuesta DRB a una red de interconexión.

Patrón	Escala?	Topología	Tamaño (nodos)	Encaminamiento	Figura
Hot-spot	Si	Toro2D	4x4 (16)	(Estático) DRB	Figura 6-23
			8x8 (64)		
	Si	Toro 3D	2x2x2 (8)		
			4x4x4 (64)		
	Si	Hipercubo	3D (8)		Figura 6-24
			4D (16)		
			6D (64)		
Butterfly	No	Toro2D	4x4 (16)		Figura 6-26
			8x8 (64)		
	No	Toro 3D	2x2x2 (8)		
			4x4x4 (64)		
	Si	Hipercubo	3D (8)		Figura 6-25
			4D (16)		
			6D (64)		

Tabla 6-8 Experimentación realizada para mostrar la escalabilidad de DRB

A continuación se muestran y comentan los resultados de cada uno de los patrones. Estos resultados se presentan con la carga normalizada respecto el ancho de bisección de la red según la propuesta del "*Workshop on Parallel Computer Routing and Communication* (PCRCW'94)" [104] para que sea equivalente la comparación de topologías de diferente tamaño. Esta forma de presentar los resultados consiste en representar la carga como una fracción de la capacidad de la red para una distribución uniforme de los destinos de los mensajes, asumiendo que los canales más altamente cargados están localizados en la bisección de la red. Esta capacidad de la red se llama *ancho de banda normalizado*. De este modo, independientemente del patrón de comunicaciones utilizado, la carga aceptada se mide como una fracción del ancho de banda normalizado. El ancho de banda normalizado se puede hallar fácilmente considerando que el 50% del tráfico uniforme generado cruza la bisección de la red. Así, si una red tiene un ancho de banda de bisección de B bits/seg., cada nodo de una red de N nodos, puede inyectar 2B/N bits/seg. como carga máxima.

6.8.1 Escalabilidad de DRB: Patrón "*hot-spot*"

La Figura 6-23 muestra los resultados del patrón de "*hot-spot*" con los métodos de encaminamiento estático y DRB para un conjunto de topologías tipo toro, variando el tamaño y la dimensión: Toro 2D 16 nodos, Toro 2D 64 nodos, Toro 3D 8 nodos y Toro 3D 64 nodos. Es necesario hacer notar que el patrón de "*hot-spot*" está "escalado" convenientemente para que represente el mismo tipo de carga en todas las topologías. Es decir, con relación a la **Figura 6-1**, que es un patrón para un toro 2D, y que tiene dos pares de mensajes (C1-C3 y C2-C4) por dirección (Norte y Sur), se añaden 2 mensajes (configurados como las demás parejas) que colisionan en el camino común por cada una de las otras dos direcciones de la tercera dimensión para el toro 3D, de manera que se aproveche el espacio tridimensional de esta topología y los 6 enlaces por nodo. Gráficamente, el patrón resultante para el caso tridimensional surgiría de rotar 90 grados el patrón existente bidimensional. Similarmente, en el caso de los hipercubos, se generan tantos mensajes que colisionan sobre un camino común como dimensiones tenga la red de interconexión respectiva.

Como puede observarse en la gráfica, todos los resultados del encaminamiento estático se agrupan en una zona mientras que los de DRB se agrupan en otra zona por debajo. La diferencia entre las respectivas curvas se incrementa a medida que la carga de tráfico crece. Si se fija un experimento (el mismo patrón, la misma carga) y se aumenta el número de nodos de la red, DRB es capaz de aprovechar al máximo la topología porque cuando la red está totalmente saturada, DRB consigue menores

incrementos de la latencia. Esto demuestra que DRB es capaz de "seguir" a la topología y el patrón, ya que cuando estos crecen, DRB sigue ofreciendo resultados similares.

Un resultado interesante es observar dos topologías del mismo número de nodos pero de diferente dimensión: el toro 3D 64 y el toro 2D 64. Se observa que los resultados con DRB son mejores en el caso 2D que el caso 3D con encaminamiento estático, por lo tanto DRB permite utilizar una red más barata con la misma capacidad de cómputo y ofrecer mejores resultados en latencia que en el caso de no utilizar ningún método de encaminamiento mejorado.

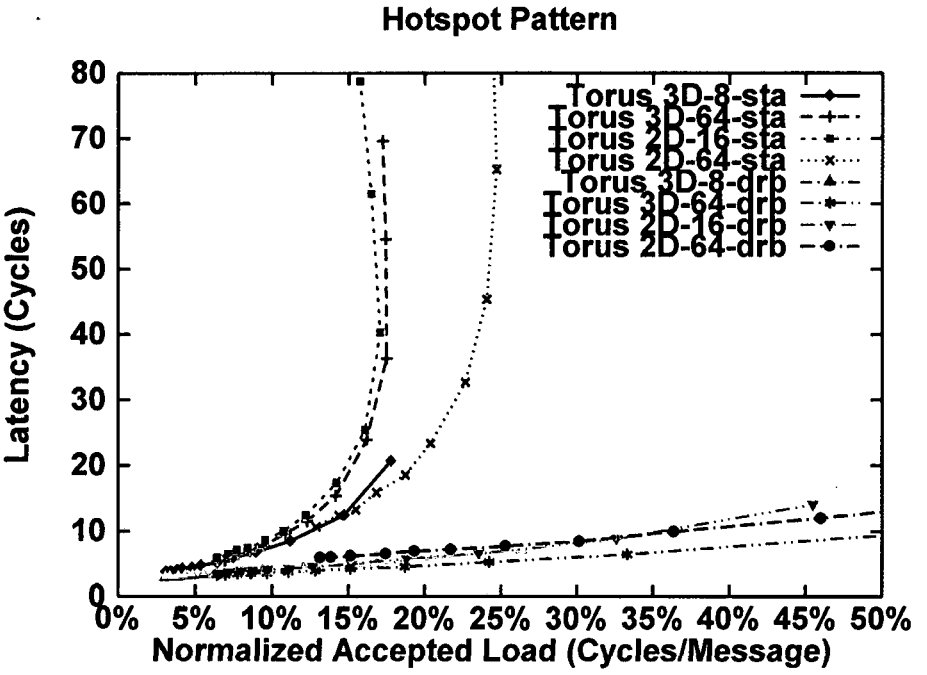


Figura 6-23 Escalabilidad de DRB para el patrón "*hot-spot*" aplicado a toros

La Figura 6-24 muestra los resultados análogos a la figura anterior, pero para diferentes topologías hipercubo: Hipercubo 3D, Hipercubo 4D e Hipercubo 6D. Como se puede observar en la Figura 6-24, los resultados para el caso estático se agrupan en una zona mientras que los resultados usando DRB se agrupan en otra zona. Esto demuestra que el patrón escala con la topología, ya que da los mismos o similares resultados con encaminamiento estático para diferentes tamaños, y que DRB también escala porque es capaz de hacer que los resultados sean similares para cualquier tamaño de topología, lo que significa que DRB es capaz de aprovechar los caminos alternativos del hipercubo a medida que la red crece en número de nodos.

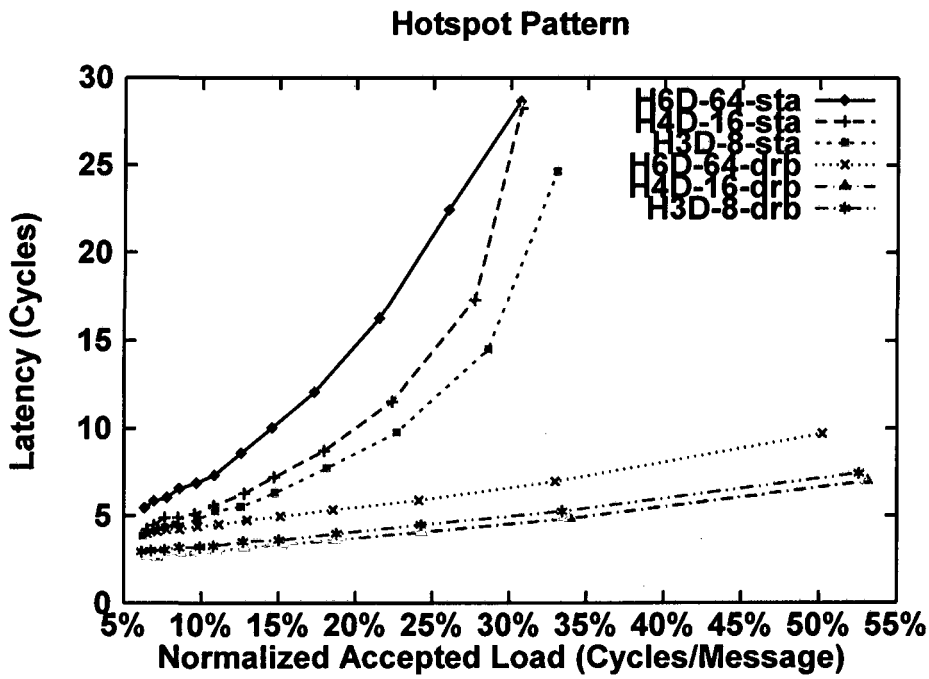


Figura 6-24 Escalabilidad de DRB para el patrón "*hot-spot*" aplicado a hipercubos

6.8.2 Escalabilidad de DRB: Patrón *"Butterfly"*

La Figura 6-25 muestra los resultados del patrón de *"Butterfly"* para diversas topologías de hipercubo. En este caso, el patrón de *"Butterfly"* escala perfectamente con la topología, es decir, que el patrón genera siempre los mismos caminos independientemente del tamaño de la red. De esta manera al normalizar la carga de entrada, los resultados de latencia obtenidos son los mismos. Nuevamente, se observa para este caso que, al escalar la topología y el patrón, DRB es capaz de escalar también sus resultados. Estos resultados son una consecuencia lógica del análisis del crecimiento de los metacaminos proporcional al de la topología realizado en el capítulo 5.

Finalmente, la Figura 6-26 muestra los resultados del patrón de *"Butterfly"* con diversos tipos de toros: Toro 2D de 16 y 64 nodos y toro 3D de 8 y 64 nodos. En este caso, el patrón no escala con la topología, es decir, al aumentar el tamaño de la red, cambia la forma de la distribución de la carga en la red, y los resultados presentan una variabilidad mayor que en los casos anteriores. En cualquier caso, con DRB siempre se consiguen mejores resultados que con el encaminamiento estático, y todos los resultados en los que se usa encaminamiento DRB están por debajo de cualquier caso estático para cualquier número de nodos y dimensión.

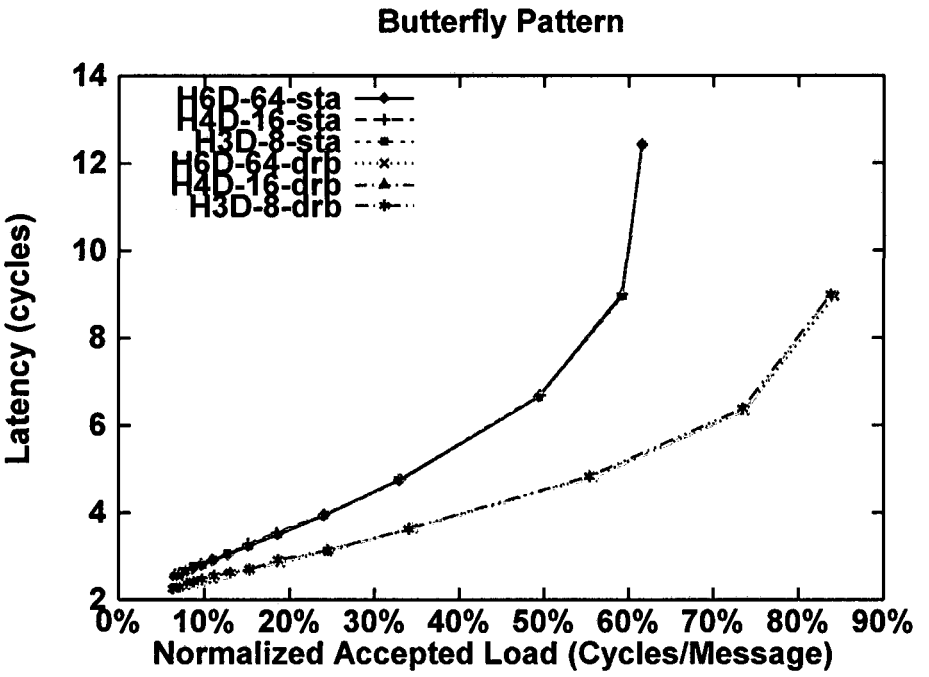


Figura 6-25 Escalabilidad de DRB para el patrón *"Butterfly"* aplicado a hipercubos

De manera similar a los casos anteriores, se observa que, para el caso de una red con 64 nodos, se permite bajar el grado de la red en una dimensión, de 3D a 2D, cuando se usa DRB, ya que se consiguen mejores resultados que en el caso de 3D con encaminamiento estático. Esto supone una ventaja muy importante de coste al disminuir el grado de los nodos, reduciendo el número de enlaces por nodo de 4 a 6, lo cual es una reducción del 50%.

Estos experimentos muestran que DRB ofrece la posibilidad al usuario final de escoger la topología que mejor se adapte a las necesidades de su aplicación. El hecho de necesitar más nodos de cómputo o enlaces de comunicación puede basarse sobre la decisión de unos valores de latencia requeridos. Usando DRB cambian totalmente las consideraciones sobre la topología a utilizar.

Tal y como se ha analizado en los experimentos, dependiendo del número de nodos, de enlaces, la carga de tráfico y los requerimientos de latencia, se puede seleccionar un toro 2D de 64 nodos con DRB en lugar de uno 3D de 64 nodos con encaminamiento estático. En la topología de hipercubo, los resultados son equivalentes por lo que se asegura al usuario que DRB será capaz de reducir la latencia de manera proporcional independientemente del número de nodos utilizados.

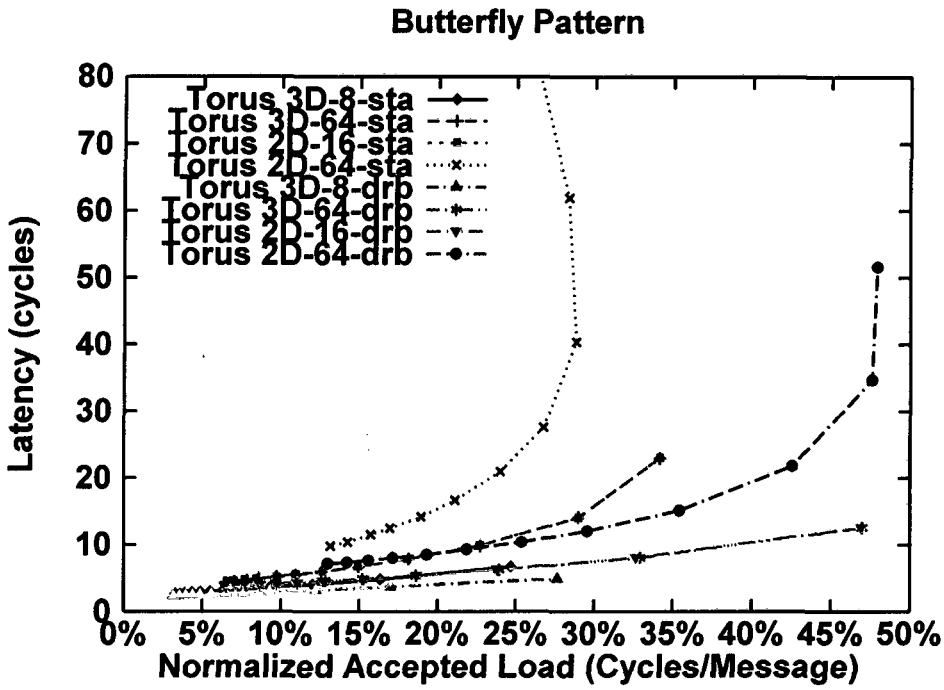


Figura 6-26 Escalabilidad de DRB para el patrón *"Butterfly"* aplicado toros

Como una conclusión general de este punto, podemos decir, a partir de los resultados representados, para las topologías analizadas, DRB es capaz de sacar partido de la capacidad de la topología y que, para un experimento escalado, DRB también escala y usa todo el ancho de banda disponible mejorando los resultados.

6.9 Conclusiones

Con este último punto, se ha finalizado la evaluación del método presentado en este trabajo de tesis. Hemos presentado una evaluación exhaustiva mostrando una serie de casos en los que se varía un aspecto de las comunicaciones mientras se mantienen otros fijos: el patrón de comunicaciones, la topología y su dimensión, el número de nodos, la carga de la red, etc.

Primeramente, se han evaluado tres aspectos específicos que hacen referencia a la información que usa DRB para configurar los metacaminos y elegir los caminos multipaso (influencia del mensaje de reconocimiento, generación temprana y retraso del mismo). La influencia del mensaje de reconocimiento sobre la carga total de la red de interconexión, nos ha mostrado que, en algunos casos, no representa una carga extra debido a que no interfiere con los mensajes de usuario y en otros casos, dependiendo del patrón de comunicaciones en la red, su influencia hace aumentar la latencia de los mensajes de usuario. En general, esta evaluación ha mostrado la nula influencia de estas cuestiones en los resultados estáticos y poca influencia en el transitorio del método

DRB. Este estudio de la respuesta transitoria y del tiempo de respuesta de DRB, en el que hemos analizado diversas alternativas respecto a la generación y retardo del mensaje de reconocimiento, ha demostrado la robustez de DRB frente a estos aspectos.

Es por ello que el resto de la experimentación se ha realizado sin utilizar la generación temprana del mensaje de reconocimiento, sin retardarlo y se ha eliminado la influencia del mismo en la red de interconexión, para obtener una cota máxima de los resultados de DRB.

A continuación, se ha realizado una experimentación más genérica en la que se ha evaluado para un conjunto de redes de interconexión (toros e hipercubos) de diversos tamaños (16 y 64 nodos) y para un conjunto de patrones estándar de comunicación (*"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"*), la respuesta en latencia, desviación estándar de la latencia y "*throughput*" para un rango de carga desde baja carga hasta saturación de tres métodos de encaminamiento: Estático, Adaptivo completo de caminos mínimos y DRB. Se ha encontrado que, para la mayoría de casos, DRB ofrece mejores prestaciones que el método adaptivo, considerado el método en la literatura que es capaz de dar los mejores resultados, y, en general, DRB mejora al caso adaptivo en un 50%, tanto en resultados de latencia como de "*throughput* ". En algunos casos, para la topología toro de 64 nodos, la latencia se reduce en una cuarta parte respecto la del adaptivo. Para el hipercubo, por las características de esta topología y de la asignación de los patrones de comunicación sobre ella las reducciones en latencia no son tan grandes, aunque de forma general, DRB supera al método adaptivo a tasas de carga máxima. Como tendencia general, se observa que la diferencia entre DRB y el encaminamiento adaptivo se incrementa al aumentar el tamaño de la red.

Es importante señalar que, aunque en esta segunda parte de la experimentación no se ha tenido en cuenta el "*overhead*" del mensaje de reconocimiento, hemos visto que las mejoras en latencia que se logran con DRB respecto a adaptativo (del orden del 50% o mayor para toros y algo menor para hipercubos) son mayores que la pérdida que supone, en el peor de los casos, incluir el mensaje de reconocimiento (del orden del 30% para toros y nula para hipercubos), con lo que DRB también supera al método adaptivo en el caso más desfavorable. De todas formas, como se ha comentado en el capítulo, si se utiliza alguna de las alternativas que eliminarían totalmente el efecto de los mensajes de reconocimiento (aprovechamiento del mensaje de sincronización del sistema de comunicaciones, reserva de un ancho de banda para los mensajes de reconocimiento, uso de una red de control separada, asignación de baja prioridad a los mensajes de reconocimiento), la ventaja de uso de DRB es grande al permitir usar la red a tasas mayores de carga aplicada.

A continuación, se ha evaluado el efecto de la longitud del paquete en *"flits"*, donde hemos observado diferencias entre unos casos y otros que no permiten generalizar. Para el patrón "*Butterfly*", la influencia es pequeña, de manera que los resultados son similares para todos los tamaños de paquete. Para el patrón "*Bit-Reversal*", el aumento del tamaño del paquete no mejora los resultados de DRB, los cuales incluso son ligeramente peores que el caso adaptivo. Para el patrón "*Perfect Shuffle*", el incremento de la longitud del paquete hace obtener a DRB mejores resultados, es decir, mayores diferencias con el adaptivo. Para el patrón "*Matrix Transpose*", DRB ofrece peores resultados que el método adaptivo a medida que se incrementa la longitud del paquete. De todos modos, ésta es una opción que siempre puede ser configurada por parte del diseñador del sistema de comunicaciones, ya que DRB no presupone ningún tamaño de paquete fijo.

Asimismo, se ha presentado una evaluación cruzada de DRB donde se ha mostrado la escalabilidad del método respecto el tamaño de la red (toros e hipercubos de 8, 16 y 64 nodos) y/o el patrón de comunicaciones ("*hot-spot*" y "*Butterfly*"). De este estudio, podemos concluir que DRB es capaz de aprovechar los recursos añadidos al incrementar el tamaño de la red y, aún más importante, que, usando DRB, es posible disminuir la dimensión de la red, con la consiguiente reducción en coste, y obtener mejores resultados que si se usa encaminamiento estático, lo cual compensaría el coste de utilizar DRB, que, como se ha puesto de manifiesto en el análisis del encaminador DRB, éste supone un coste lineal con el número de nodos de la red. Como se ha visto, DRB es capaz de aprovechar el incremento en tamaño de las redes, tanto de toros como hipercubos, aunque la topología toro es más fácilmente ampliable que el hipercubo, ya que se mantiene de grado constante, mientras que para el hipercubo es necesario aumentar el número de enlaces de todos los nodos al aumentar el número de ellos.

Sobre la base de todo esto, como conclusión general, puede deducirse que DRB es una buena alternativa como método general de encaminamiento de mensajes en una red de interconexión de altas prestaciones que opere bajo un amplio rango de situaciones.

Capítulo 7 Conclusiones y líneas abiertas

7.1 Conclusiones

Con este capítulo, llegamos al final de la presente memoria de tesis que expone todo el trabajo realizado sobre las redes de interconexión de computadores de altas prestaciones paralelos. A lo largo de la memoria, hemos visto en una serie de capítulos desde el punto inicial de exposición de la motivación del tema de trabajo elegido en torno a las redes de interconexión y de la presentación del marco de las ciencias de la computación de altas prestaciones actual hasta la propuesta de mecanismo de balanceo de las comunicaciones, su definición, análisis y evaluación.

Partimos de la observación del funcionamiento de las redes de interconexión en patrones típicos de comunicación que aparecen en aplicaciones paralelas en los campos de la matemática, la ciencia y la tecnología. Esta observación nos dice que la saturación

se produce a tasas bajas de carga de la red (menos del 50% de capacidad de carga) y aparece de manera súbita en forma de recta vertical con un cambio muy repentino.

A lo largo de este camino, hemos establecido los objetivos planteados para el trabajo, centrados en realizar aportaciones al diseño de las redes de interconexión para los computadores de altas prestaciones que contribuyan a la definición de la arquitectura de dichos computadores. A partir de aquí, hemos realizado un estudio del modelado de las redes de interconexión presentando dos modelos diferentes. El primero es un modelo analítico del comportamiento dinámico de las redes de interconexión y el segundo es un modelo funcional que simula el comportamiento de los encaminadores de las redes de interconexión. El modelo analítico ha sido utilizado para analizar el comportamiento de las redes de interconexión. Este modelo nos ha permitido realizar un análisis y adquirir una comprensión del comportamiento de la latencia en las redes de interconexión. El modelo funcional ha servido para realizar una evaluación exhaustiva de las propuestas de encaminamiento realizadas en este trabajo de tesis.

Del análisis del comportamiento de las redes de interconexión, hemos puesto de manifiesto una serie de problemas que surgen en el uso de redes de interconexión y hemos extraído sus causas, centradas en una no-coincidencia de la distribución de la carga de comunicaciones presente en la red de interconexión con la topología de la red.

Esta problemática puede resumirse en dos aspectos principales. La respuesta no lineal, sino más bien exponencial, de la latencia frente al incremento de la carga de comunicaciones a partir de un cierto valor umbral, por un lado, y el efecto de rápida propagación de la saturación de las comunicaciones en la red de interconexión desde puntos focalizados a toda la red de interconexión.

Este comportamiento es debido a la aparición de "*hot-spots*" y sus características ya mencionadas: efecto "dominó" (multiplicativo) y propagación muy rápida.

A continuación, hemos definido cuál debería ser el comportamiento ideal de una red de interconexión para que ofreciese un alto rendimiento y facilitase la tarea de programación del computador paralelo. Hemos resumido este comportamiento en que la red debería ofrecer una latencia baja y uniforme en un amplio rango de carga o patrón de comunicaciones. Ofreciendo este requerimiento, se tendría una visión de la red de interconexión que ofrece una conexión de todos con todos entre los procesadores y a la misma distancia (o con el mismo ancho de banda), incluso para "cualquier" carga presente en la red. En nuestro caso, no homogeneizamos la latencia en vacío, sino que en presencia de carga en un amplio rango de valores es cuando nos mantenemos en valores de latencia que no crecen exponencialmente.

Con el objetivo de latencia uniforme se pretende aprovechar al máximo el ancho de banda de toda la red durante el mayor rango de carga de comunicaciones aplicado posible. Si la latencia es uniforme, hemos visto a lo largo del trabajo, que la granularidad de la aplicación será válida para un rango más amplio de carga de la red. Por otro lado, con una latencia uniforme y predecible, la asignación de procesos a procesadores ("*mapping*") se simplifica porque se pueden tener en cuenta las comunicaciones de manera sencilla.

A partir de la definición de los objetivos y del análisis del comportamiento de las redes de interconexión realizado anteriormente, hemos introducido cuál debería ser el tipo de solución necesaria y hemos definido el concepto de balanceo del tráfico para conseguir un uso uniforme del ancho de banda de la red y eliminar los "*hot-spots*". Esta solución consiste, según nuestra propuesta, en el balanceo de la carga de comunicaciones en la red de interconexión, por lo que hemos presentado el mecanismo introducido en este trabajo para conseguir los objetivos propuestos consistente en el Balanceo Distribuido del Encaminamiento o DRB por sus siglas en inglés ("*Distributed Routing Balancing*"). La técnica del balanceo se basa en la distribución del tráfico usando nuevos caminos alternativos. Es un método dinámico que usa información del comportamiento de la red obtenida a un nivel local.

Este mecanismo se basa en la expansión de los caminos controlada por la carga de comunicaciones. Con este método se pretende conseguir una uniformización de la latencia, lo que es a su vez un método de eliminar los "*hot-spots*" y evitar la contención de mensajes. El método de DRB pretende desacoplar el patrón de tráfico de la aplicación de la topología física de la red de interconexión.

Al igual que el Balanceo del Cómputo ("*Dynamic Load Balancing*") en que se mueven "procesos" de un procesador a otro, DRB pretende realizar un Balanceo de la Carga de Comunicaciones ("*Communication Load Balancing*") para mover unos flujos de mensajes de unos enlaces físicos a otros. Empezando a partir de los trabajos de Valiant y May, ya mencionados en esta memoria, sobre "*Random Routing*", DRB busca crear nuevos caminos alternativos entre cada par fuente-destino par balancear las comunicaciones.

Como se ha dicho, este balanceo pretende que la latencia este controlada y sea uniforme, lo que permite que sea predecible y aplicar técnicas de solapamiento de cómputo y comunicaciones para ocultar la latencia. Con ello, se permite un uso mayor de la red de interconexión, porque la saturación se produce a un nivel de carga superior.

La técnica de DRB se divide en dos partes para crear y utilizar los nuevos caminos alternativos. La primera parte de DRB se basa en la definición de tres entidades: Supernodos, Caminos Multipaso y Metacaminos. La segunda parte es una política que hace referencia a cómo se utilizan las entidades definidas en la primera parte. Esta política se divide en tres partes: Monitorización de la Latencia, Configuración de Metacaminos y Selección de Caminos Multipaso.

También hemos remarcado la importancia del efecto colectivo en toda la red de interconexión de cooperación y de ajuste mediante la interacción entre todos los canales de la red de interconexión.

La introducción de la técnica DRB en una red de interconexión implica, de hecho, el diseño de un encaminador físico. El diseño de tal encaminador DRB se ha analizado y se ha cuantificado su coste tanto temporal como espacial.

La siguiente parte de la memoria se ha dedicado a evaluar extensivamente esta nueva propuesta introducida. Primeramente, se han evaluado las características estáticas que el método ofrece, lo que representa una medida "en vacío" de la capacidad del método. Con ello se ha evaluado el efecto que produce tener un mayor grado de paralelismo al agregar un mayor número de procesadores, lo que permite una menor granularidad del programa de aplicación.

Finalmente, hemos realizado y analizado una evaluación exhaustiva de múltiples aspectos de DRB, tanto cuantitativos como cualitativos. Para evaluar el método DRB en presencia de tráfico real y compararlo frente a otras alternativas estáticas y adaptivas, hemos utilizado la herramienta de simulación de redes de interconexión desarrollada en este trabajo de tesis.

Se ha experimentado con patrones sintéticos, porque representan los patrones de comunicación que aparecen en las aplicaciones paralelas más comunes, para obtener la respuesta en latencia del método y también se han analizado aspectos como el tiempo de respuesta de DRB, analizando dos variantes de la política DRB, o la influencia del mensaje de reconocimiento.

Con todo ello, se espera haber contribuido a definir las características de los computadores paralelos o de altas prestaciones del futuro.

En este punto, queremos recordar los principales objetivos de este trabajo y las principales aportaciones conseguidas:

- ✓ Se ha realizado un modelo analítico de las redes de interconexión que representa el comportamiento temporal de la latencia de los mensajes en la red. Se basa en la deducción de una serie de ecuaciones estocásticas que representan las colisiones en la red de interconexión. Este modelo es capaz de representar cualquier patrón sobre cualquier topología de red de interconexión.

- ✓ Se ha realizado el estudio y confección de un simulador funcional que simule el comportamiento de las redes de interconexión. Este simulador permite obtener resultados tanto estacionarios como transitorios del comportamiento en latencia de cualquier patrón de comunicaciones sobre cualquier topología e incorpora los métodos de encaminamiento estático, completamente adaptativo y todos los métodos de DRB desarrollados en esta tesis.

- ✓ Se han estudiado las características dinámicas de las redes de interconexión, mediante una serie de ejemplos básicos, poniendo de manifiesto las problemáticas de funcionamiento y los parámetros de la carga de la red que las provocan. Se ha observado que el número de mensajes, la frecuencia de inyección o la longitud de los mensajes son factores clave a la hora de saturar la red de interconexión y producir "*hot-spots*".

- ✓ Se ha realizado una propuesta de comportamiento deseable de una red de interconexión, a partir de su respuesta en comportamiento de la latencia y el "*throughput*", centrada en ofrecer una latencia controlada e uniforme. La importancia de la latencia uniforme es, que de esta manera, se puede predecir o aproximar los tiempos de comunicación de las tareas en ejecución. Con esta información es fácil hacer una asignación de tareas a nodos de procesamiento que optimice la utilización de los procesadores y minimice el tiempo de ejecución del programa paralelo.

- ✓ Se ha introducido el "*balanceo dinámico de las comunicaciones*" como la técnica base para conseguir el comportamiento deseado de las redes de interconexión. Esta técnica es paralela al balanceo de la carga de computación entre los nodos de procesamiento y su objetivo es distribuir uniformemente la carga de comunicaciones entre todos los recursos disponibles de la red de interconexión.

- ✓ Se ha descrito la propuesta del mecanismo de encaminamiento para conseguir el comportamiento establecido: El *Balanceo Distribuido del Encaminamiento*. Este mecanismo propone distribuir las comunicaciones mediante la expansión a caminos alternativos de los caminos estáticos usados en la red de interconexión. Esta expansión es dinámica para cada par fuente-destino y esta controlada por la latencia

sufrida por los mensajes enviados. Para la creación de los caminos alternativos se utiliza la técnica de los destinos intermedios. DRB define cómo crear y utilizar los destinos intermedios para distribuir las comunicaciones en la red de interconexión.

- ✓ Se ha realizado la caracterización de las redes de interconexión respecto la distancia promedio, desviación de la distancia y ancho de banda cuando se utiliza DRB. Esta simulación ha supuesto la caracterización de los metacaminos definidos por DRB y de ella se ha observado que aunque se produce un incremento de la longitud del camino recorrido por los mensajes, es mayor el incremento que se produce en el ancho de banda (número de nodos y por tanto, de enlaces disponibles para realizar la comunicación) es todavía mayor. Se han estudiado todas estas características dependiendo de la topología y su tamaño. Finalmente, se ha observado que este comportamiento se mantiene si se aumenta el tamaño de la red, lo cual demuestra una buena escalabilidad de DRB, lo que asegura que los resultados se mantienen al aumentar el número de nodos de la red.

- ✓ Se ha realizado el estudio y análisis de las propuestas introducidas mediante la comparación vía simulación con las técnicas estática (tomada como base a mejorar) y completamente adaptativa (tomada como la técnica que ofrece mejores prestaciones en la literatura), donde se ha mostrado la idoneidad de la propuesta frente a un numeroso grupo de casos.

  Primeramente, se han evaluado tres aspectos específicos que hacen referencia a la información que usa DRB para configurar los metacaminos y elegir los caminos multipaso (influencia del mensaje de reconocimiento, generación temprana y retraso del mismo). Este estudio de la respuesta transitoria y del tiempo de respuesta de DRB, en el que hemos analizado diversas alternativas respecto a la generación y retardo del mensaje de reconocimiento, ha demostrado la robustez de DRB frente a estos aspectos. Se ha puesto de manifiesto la no penalización de la monitorización de DRB y que las ventajas de utilizar DRB compensan el "overhead" que pueda introducir la monitorización.

  A continuación, se ha realizado una experimentación más genérica en la que se ha evaluado para un conjunto de redes de interconexión (toros e hipercubos) de diversos tamaños (16 y 64 nodos) y para un conjunto de patrones estándar de comunicación (*"Butterfly"*, *"Bit-Reversal"*, *"Perfect Shuffle"* y *"Matrix Transpose"*), la respuesta en latencia, desviación estándar de la latencia y "*throughput*". Se ha encontrado que, para la mayoría de casos, DRB ofrece mejores prestaciones que el método adaptivo, considerado el método en la

literatura que es capaz de dar los mejores resultados, y, en general, DRB mejora al caso adaptivo en un 50%, tanto en resultados de latencia como de "*throughput* ". Esto significa que DRB permite hacer un uso de la red a tasas mayores de carga que otros métodos sin que se llegue a la saturación, lo cual significa que pone a disposición un mayor ancho de banda para ser utilizado por los mensajes de usuario. Este incremento de uso de la red compensa el coste añadido de monitorización que realiza DRB.

También, se ha evaluado la influencia de la longitud del paquete y se ha encontrado que, para algunos patrones, el incremento de la longitud del paquete implica una mejora de los resultados mientras que para otros no es así.

Finalmente, se ha realizado un estudio sobre la escalabilidad de DRB, a partir del cual podemos concluir que DRB es capaz de aprovechar los recursos añadidos al incrementar el tamaño de la red y, aún más importante, que, usando DRB, es posible disminuir la dimensión de la red, con la consiguiente reducción en coste, y obtener mejores resultados que si se usa encaminamiento estático, lo cual compensaría el "*overhead*" de utilizar los encaminadores DRB.

Todos estos resultados demuestran la validez del método DRB cómo método de encaminamiento de mensajes en redes de interconexión de computadores paralelos de propósito general, el cual es capaz de reducir la latencia y aumentar el rango de carga en el cual la red es operativa.

Finalmente, queremos remarcar que el presente trabajo ha sido evaluado y presentado en las diversas conferencias:

I. Garcés, D. Franco, E. Luque "Improving Parallel Computer Communication: Dynamic Routing Balancing" Euromicro Workshop on Parallel and Distributed Processing (PDP-98) IEEE CS Press. USA, 1998 pp.111-119

D.Franco, I.Garcés, E. Luque "Dynamic Routing Balancing in Parallel Computer Interconnection Networks" Vector and Parallel Processing- Vecpar'98. Selected Papers. . Lecture Notes on Computer Sciences (LNCS 1573) Springer Verlag. 1999 pp.494-507

D.Franco, I.Garcés , E. Luque "Balanceo del Encaminamiento Distribuido para la Comunicación en Redes de Interconexión" IV Congreso Argentino en Ciencias de la Computación. CACIC98 Argentina (1998) pp. 665-680

D Franco, I Garcés , E Luque "Distributed Routing Balancing for Interconnection Networks" Procc. of Intl. Conf. on High Perf. Compt. (HiPC98). IEEE Computer Society pp. 253-261

D Franco, I Garcés, E Luque "Avoiding Communication Hot-Spots in Interconnection Networks" Procc. of 32nd Hawaii Intl. Conf. on System Sciences. (HICSS99) IEEE Society. CD-ROM Proceedings.

D Franco, I Garcés, E Luque "A new method to make communication latency uniform:". Procc. of ACM International Conference on Supercomputing (ICS99) pp.210-219.

D Franco, I Garcés, E Luque "Analytical Modelling of the Network Traffic Performance" Proc. Modelling, Analysis and Simulation of Computers and Telecommunicacion Systems (MASCOTS99). IEEE Computer Society . Oct 1999 pp. 190-196

7.2 Líneas abiertas

Este trabajo, que ha presentado un método completo de balanceo de las comunicaciones en la red de interconexión, deja una serie de líneas abiertas abordables como trabajo futuro. Entre ellas podemos citar:

7.2.1 Variantes del método DRB

Como se ha visto en el capítulo 4, DRB se basa en dos componentes, la definición de caminos alternativos y la selección de dichos caminos en función de la carga presente. Hasta ahora hemos descrito una posible definición e implementación concreta de estos aspectos.

Para la creación de caminos alternativos hemos desarrollado los conceptos de Supernodo, metacamino y camino multipaso. Para la selección de un camino multipaso hemos diseñado una política basada en tres fases. Estas fases son monitorización e información del estado de la red, configuración de nuevos metacaminos y selección de un camino multipaso.

En este punto se van a comentar posibles variantes a esos puntos que dan como resultado un conjunto de posibilidades para cada uno de los aspectos. Lo que no se modifican son los conceptos básicos de DRB por lo que respecta al balanceo de las comunicaciones y a la consecución de ese objetivo mediante la búsqueda de caminos alternativos. Estos conceptos de base se muestran en la Figura 7-1, ya introducida en el

capítulo 4. Ahora comentamos varias posibles alternativas sobre cómo formar los caminos alternativos y sobre las fases del encaminamiento DRB.

A continuación se comentan las posibles alternativas para cada uno de los diferentes aspectos.

7.2.1.1 Definición de caminos alternativos

Respecto la definición de caminos alternativos, proponemos que se realice mediante la selección de destinos intermedios. Lo que puede variarse es el cómo se seleccionan los destinos intermedios. Esta selección puede basarse en otros principios diferentes a los Supernodos. Un ejemplo sería definir y buscar un tipo de "mediatriz" del segmento formado entre los nodos fuente y destino y utilizar la mediatriz como Supernodo único.

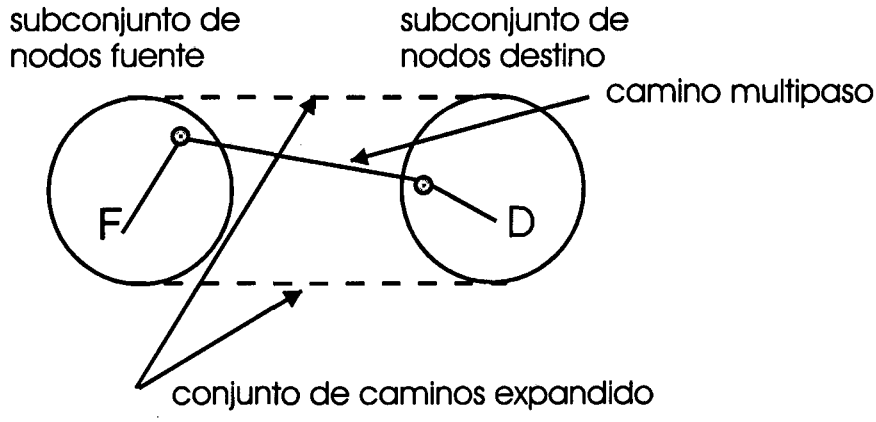


Figura 7-1 Concepto de DRB

Otro aspecto a considerar es que, sea cual sea el modo que se seleccionan los caminos multipaso, éstos se configuren de manera que, para un mismo fuente y destino, todos los caminos multipaso sean totalmente disjuntos. Es decir, que no compartan ningún nodo en común. Para ello se podrían utilizar modelos de tablas previas que configuran los caminos multipaso de una red. Esta alternativa puede tener ventajas o desventajas frente a la propuesta con respecto al número de caminos disponibles y el aprovechamiento de cada camino.

7.2.1.2 Encaminamiento DRB

Respecto al encaminamiento DRB, actualmente la monitorización se hace recogiendo un único valor de total de la latencia sufrida a lo largo de todo el camino e informando de este valor al nodo fuente cuando se llega al final del recorrido. Ésta es la

aproximación más simple, pero no es la única posible, ya que son posibles varias alternativas tanto sobre la cantidad de información que se registra y sobre cuándo se informa de esa información.

Asimismo, la política de configuración de metacaminos presentada, actúa sobre el valor del ancho de banda instantáneo del metacamino actual para configurar un nuevo metacamino. Nuevamente, varias alternativas sobre la cantidad de información y la antigüedad de esa información son posibles.

A continuación, se presentan las siguientes alternativas sobre cada aspecto mencionado.

a) Monitorización

Respecto la monitorización, hay dos posibilidades de actuación. En el registro de la latencia y sobre el informe de la latencia. Los diferentes diseños alternativos de DRB respecto el informe de la latencia ya fueron introducidos y desarrollados en el capítulo 4 y la experimentación de ellos respecto el tiempo de respuesta de DRB se han presentado en el capítulo 6, así que aquí se comentarán las alternativas respecto el registro de la latencia.

En lugar de registrar el total de latencia sufrida durante todo el camino con un único valor, se puede registrar la latencia por tramos para obtener mas información. A este respecto, podríamos considerar tres alternativas posibles que siguen una gradación desde una única latencia hasta máxima información y que son: primera, almacenar una latencia para cada uno de los pasos del camino multipaso, segunda, almacenar la latencia para cada una de las dimensiones que recorre el mensaje, en el caso de una red regular tipo n-cubo k-ario, y tercera, almacenar una latencia por cada enlace que atraviesa el mensaje.

Evidentemente, cada uno de los sistemas que almacena más información ofrece por un lado más posibilidades para conocer exactamente dónde se encuentran los problemas de la red de interconexión y poder tomar decisiones más inteligentes, pero por el otro, mayor espacio se necesita para ellas. Se debería buscar un equilibrio entre cantidad de información, utilidad y espacio necesario. Por ello, alternativas de las cuales no es posible conocer la cantidad de espacio requerido, como la última, no serían factibles, mientras que el registro de la latencia por pasos del camino multipaso ofrece mucha información con sólo un número de valores finito.

b) Configuración de los metacaminos

Sobre el algoritmo de configuración de metacaminos, se pueden dar varias alternativas. Aquí tenemos varios aspectos a considerar: el primero es sobre qué información se activa el algoritmo, qué información utiliza para configurar los supernodos y cómo actúa sobre los supernodos. Actualmente, la versión básica actúa sobre el ancho de banda instantáneo del metacamino y, en función de este valor modifica en una o varias unidades los tamaños de los supernodos.

Las alternativas serían, por un lado, en lugar de actuar sobre el ancho de banda instantáneo del metacamino, guardar un histórico de una cierta cantidad de valores sobre un registro de desplazamiento, y utilizar esa información para descubrir tendencias de la latencia, para hacer decisiones más inteligentes. Asimismo, podría considerarse no sólo la latencia del metacamino completo, sino los valores de latencia individuales de cada uno de los caminos multipaso que forman el metacamino.

Por otro lado, con respecto al dimensionamiento de los supernodos, podría actuarse incrementándolos o decrementándolos de una en una unidad en lugar de en varias de golpe. Si se usa la alternativa de comunicar la latencia durante el viaje del mensaje, y no al final, puede ser conveniente aumentar el tamaño de los supernodos de uno en uno para tener un incremento de caminos multipaso gradual, según va llegando la información de latencia.

Finalmente, se pueden configurar supernodos no regulares, con un conjunto de nodos arbitrario en función de la información de latencia recibida.

c) Selección de los caminos multipaso

Para esta fase del encaminamiento DRB, las alternativas de diseño irían en función de las alternativas de la fase de monitorización. La distribución de los mensajes siempre se haría entre los caminos multipaso del metacamino de manera proporcional a la latencia de cada uno de ellos.

En el caso de que se utilizase un método adelantado de información de latencia en la fase de monitorización, ésta información sólo se utilizaría para la configuración de los metacaminos y no para la selección de los caminos multipaso. Para esta última tarea hace falta, por razones obvias, disponer de toda la información de latencia de los caminos multipaso en cuestión. Es por ello que la información de latencia total desde el nodo destino al origen siempre debe producirse.

Por otro lado, en el caso de que la información de la latencia total de los caminos multipaso se disponga de manera dividida en pasos, dimensiones o enlaces, la selección puede hacerse teniendo toda esa información en cuenta.

Con todas estas alternativas, se tiene un espacio de diseño multidimensional complejo donde unas encajarían mejor que otras y donde debería buscarse un compromiso entre prestaciones y coste.

Los diferentes diseños alternativos de DRB pueden dar lugar a variaciones en el comportamiento. Un factor clave de DRB es el tiempo de respuesta, es decir, el tiempo desde que se detecta una situación de saturación hasta que se actúa para corregirla. Evidentemente, cuanto menor sea este tiempo, mejor será la respuesta conseguida con DRB. Incluso si se pueden adelantar las decisiones, previniendo las situaciones problemáticas en función de ciertas tendencias de comportamiento de la latencia, será mucho mejor. En este sentido, las alternativas de adelantar el informe de la latencia en cuanto supera un cierto umbral o actuando en función de un histórico de la latencia, pueden mejorar el comportamiento de DRB.

La profundización en las diferentes variantes de DRB aquí presentadas. Este punto pasaría por la implementación y evaluación bajo determinados casos prueba de cada una de las alternativas expuestas con objeto de compararlas entre ellas y encontrar la mejor o establecer los criterios de coste y rendimiento que determinan cuál es la mejor de las alternativas dependiendo de las circunstancias presentes en la red de interconexión y el programa de aplicación aplicado.

7.2.2 Migración de procesos

La inclusión de la migración de procesos debido a causas de comunicaciones. En relación con este punto, hemos ideado una extensión para trabajar a dos niveles en DRB con objeto de crear los caminos alternativos para balancear el tráfico en la red. El primer nivel es el método presentado en este trabajo de tesis de creación de Supernodos y Metacaminos y el uso de caminos multipaso. Este nivel es un nivel internodo, el cual cambia los caminos de los mensajes pero sin mover las posiciones de los nodos fuente o destino. El segundo nivel es un nivel intranodo el cual mueve procesos por cuestiones de carga de comunicación. El nivel intranodo se ha diseñado para el siguiente caso. Supóngase que un único nodo presenta un requerimiento de ancho de banda para la inyección o recepción de mensajes por parte de sus procesos internos mayor que lo que sus enlaces físicos pueden aceptar. En este caso, la distribución de caminos realizada por DRB no puede mejorar la situación de ninguna manera y entonces, se deben mover los procesos que generan esos mensajes. Si esta situación de saturación es provocada

por varios procesos cuya suma del ancho de banda necesario es mayor que la de los enlaces físicos, entonces, DRB intentará mover uno o más procesos a otro procesador para distribuir las comunicaciones y evitar esta situación. Pero, si los requerimientos de comunicación vienen de un único proceso, entonces, la migración del proceso no puede solucionar el problema y, en este caso se enviara un mensaje de aviso al usuario informando de la situación y aconsejándole que cambie la aplicación porque su granularidad no es adecuada en ese entorno de ejecución. Esta situación local de saturación relativa al nivel intranodo es de fácil detección mediante el examen de las colas de inyección del sistema de comunicaciones, en el caso de procesos emisores, o, en el caso de procesos receptores, los vecinos más próximos informarían a un nodo de que sus colas están llenas de paquetes dirigidos a ese nodo. Como se puede observar, este nivel intranodo tiene en cuenta los objetos de la aplicación del usuario (procesos y canales) y vuelve a ir en la dirección de facilitar la tarea de diseño y programación de las aplicaciones paralelas por parte del usuario, detectando una granularidad inadecuada cuando la distribución de caminos de DRB no puede arreglar esa situación.

7.2.3 DRB y calidad de servicio (QoS)

La modificación de DRB para conseguir cumplir una serie de requerimientos o calidades del servicio (QoS) capaz de dar respuesta a las necesidades de aplicaciones multimedia. En este sentido, el diseño actual de DRB está adaptado a las características de las aplicaciones paralelas de cómputo intensivo. Tanto en el aspecto en cómo se modela la aplicación paralela como en la respuesta en latencia que se espera obtener. Por un lado la aplicación se modela, tal y como se ha descrito, como una serie de envíos de mensajes separados por tiempos de cómputo independientes unos de los otros. Por otro lado, se intenta balancear las comunicaciones para conseguir una latencia mínima y uniforme en un esquema tipo "*best-effort*" en el que se intenta obtener los mejores resultados posibles. Las aplicaciones de tipo multimedia, como se comentó en la introducción de esta memoria, no se ajustan a ese modelo ni precisan una respuesta de ese tipo. Este punto incluiría, tanto la descripción y modelización de las aplicaciones multimedia como la definición del tipo de respuesta en latencia esperado. En el primer punto se deben incluir cuestiones como la generación de paquetes de datos de forma periódica, la tasa máxima y mínima de generación, etc. En el segundo punto se deben considerar aspectos como la latencia máxima tolerada, la cadencia de llegada de paquetes, etc. Todos estos aspectos son importantes a la hora de considerar un sistema multimedia como la distribución de vídeo bajo demanda, por ejemplo.

7.2.4 DRB y tolerancia a fallos

Aplicar la multiplicidad de caminos de DRB a asegurar una cierta tolerancia a fallos. DRB, al usar varios caminos para enviar los mensajes entre un nodo fuente y un nodo destino, puede ser robusto ante el fallo de algún enlace en la red de interconexión. La detección de un camino con fallos puede hacerse aprovechando los propios mensajes de reconocimiento de DRB y considerando tiempos máximos de espera para considerar un camino inutilizado. Los caminos alternativos se pueden utilizar para reenviar mensajes que no han alcanzado el destino.

7.2.5 Aplicación a redes irregulares

Esta línea abierta correspondería a aplicar el método DRB a redes irregulares. Frente a las redes regulares simétricas altamente utilizadas en supercomputadores paralelos, las redes irregulares son una alternativa cuando se consideran "*Networks of Workstations*" o redes basadas en tecnologías como Myrinet. DRB es perfectamente aplicable a este tipo de redes, ya que no supone ninguna configuración específica de la topología. En este caso es importante considerar la multiplicidad de caminos disponible, es decir, cuántos caminos proporciona la red entre cada par de nodos.

7.2.6 Guía de uso de DRB

Elaborar una "guía de uso" de DRB, presentando una serie de casos tipo. Esta guía ofrecería la respuesta en latencia cuando se usa el método de DRB con diferentes Metacaminos para diferentes casos tipo de colisión de canales. Con esta guía el usuario podría saber, a partir de los canales de la aplicación y la latencia que quiere obtener, cuántos caminos múltiples deberá utilizar. Con ello, se puede estudiar los requerimientos que DRB impone a la red de interconexión para asegurar una cierta latencia.

BIBLIOGRAFÍA

[1] VS Adve, MK Vernon "Performance analysis of multicomputer mesh interconnection networks with deterministic routing" IEEE trans. Parallel Distr. Systems 5, 1994 pp.

[2] A Agarwal "Limits on Interconnection Network Performance" IEEE Trans. on Parallel and Distributed Systems, Vol. 2 No.4 Oct. 1991 pp. 398-412

[3] DP Agrawal, VK Janakiram, GC Pathak "Evaluating the Performance of Multicomputer Configuration" IEEE Computer Vol. 19 No. 5 1986 pp. 23-39

[4] S Akl "The Design and Analysis of Parallel Algorithms" Prentice-Hall 1989.

[5] TE Anderson, DE Culler, DA Patterson et al. "A Case for NOW (Networks of Workstations)" IEEE Micro 15(1) 1995 pp.54-64

[6] A Arruabarrena "Análisis y Evaluación de Sistemas de Interconexión para procesadores Masivamente Paralelos" Tesis Doctoral, Universidad del País Vasco. Sept. 1993

[7] WC Athas, CL Seitz "Multicomputers: Message-Passing Concurrent Computers" IEEE Computer, v.21, n.8 1998 pp.9-24

[8] J Banks, JS Carson, B Nelson "Discrete-Event System Simulation" Prentice-Hall, Inc. 1996.

[9] P Baran "On distributed communications networks" IEEE Trans. On Commun. Systems Vol. C S-12 Mar. 1964 pp. 1-9

[10] GH Barnes, RM Brown, M Kato, DJ Kuck "The ILLIAC IV Computer" IEEE Transactions on Computers C-17(2) , 1968 pp. 746-757

[11] J Beecroft, M Homewood, M McLaren "Meiko CS-2 interconnected Elan-Elite design" Parallel Computing, V.20, No.10-11, Nov. 1994, pp.1627-1638

[12] R Beivide, E Herrada, JL Balcázar, A Arruabarrena "Optimal Distance Networks of Low Degree for Parallel Computers" IEEE Trans. on Computers. Vol. 40. No. 10. Oct 1992, pp. 1109-1124.

[13] DP Bertsekas, JN Tsitsiklis "Parallel and Distributed Computation. Numerical Methods" Cap. 5Prentice-Hall, 1989.

[14] Boden et al "Myrinet: A Gigabit-per-second Local Area Network" IEEE Micro 15(1) 1995 pp. 59-64

[15] L Bhuyan "Interconnection Networks for Parallel and Distributed Processing" IEEE Computer June 1987 pp. 9 - 12.

[16] Bokhari, Shahid. " Multiphase Complete Exchange on Paragon, SP2, and CS-2" IEEE Parallel & Distributed Technology, Vol. 4, No.3, Fall 1996, pp. 45-59.

[17] K Bolding "Chaotic Routing: Design and Implementation of an Adaptive Multicomputer Network Router" PhD Thesis, University of Washington, Department of Computer Science and Engineering, Seattle, WA, July 1993

[18] K Bolding, M Fulgham, L Snyder "The Case of Chaotic Adaptive Routing" IEEE Trans. On Computers, Vol. 46, n. 12, Dec 1997, pp.1281-1292,

[19] J Carbonaro, F Verhoorn "Cavallino: The Teraflops Router an NIC" Procc Hot Interconnects Symp. IV Aug 1996, pp. 157--160

[20] U Carlini, U Villano "The routing problem in transputer-based parallel systems" Microprocessors and Microsystems, Vol. 15 No. 1 Jan-Feb 1991. pp. 21-33

[21] C Carrión, R Beivide, JA Gregorio, F Vallejo "A flow control mechanism to avoid message deadlock in k-ary n-cubes" Intl Conf. High Perf. Computing, Dec 1997, pp. 322-329

[22] C Carrión, JA Gregorio, JM Prellezo, R Beivide, R Menéndez "Limitaciones de las Herramientas de Simulación para Redes de Interconexión" VII Jornadas de Paralelismo 1996 pp. 291-307

[23] C Carrión, C Izu, JA Gregorio, F Vallejo, R Beivide "Ghost Packets: A deadlock free solution for k-ary n-cube networks" 6th Euromicro Workshop on Parallel and Distributed Processing Jan 1998 pp. 22-32,

[24] D Chaiken, C Fields, K Kuvihara, A Agarwal "Directory-Based Cache Coherence in Large Scales Multiprocessors" IEEE Computers, Vol. 23, No. 6, June 1990, pp 49-58.

[25] KM Chandy, J Misra. "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs" IEEE Transactions on Software Engineering, Vol. SE-5 No. 5 Sept. 1979, pp. 440-452

[26] AA Chien, JH Kim "Planar Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors" Journal of the ACM, 42(1) , Jan. 95 pp. 91-123

[27] C Clos "Study of Non-Blocking Switching Networks" The Bell Systems Technical Journal, 1953, pp.406-424

[28] Cray Research Inc. "Cray T3D System Architecture Overview". 1st Ed. Sep 1993.

[29] DE Culler, JP Singh "Parallel Computer Architecture. A Hardware/Software Approach" Morgan Kaufmann Pub. 1999.

[30] WJ Dally "A VLSI Architecture for Concurrent Data Structures". Kluwer Academic Press, Hingham, MA, 1987

[31] WJ Dally, CL Seitz "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks" IEEE Transactions on Computers, Vol. C-36, No.5, Mayo 1987, pp. 547-553.

[32] WJ Dally "Performance Analysis of k-ary n-cube Interconnection Networks". IEEE Transactions on Computers, Vol. 39, No.6, Jun 1990, pp. 775-785.

[33] WJ Dally "Network and Processor Architectures for Message Driven Multicomputers" VLSI and Parallel Computation. Morgan Kaufmann Publishers, 1990, pp. 140-222

[34] SP Dandamudi, DL Eager "Hot-Spot Contention in Binary Hypercube Networks" IEEE Transactions on Computers, Vol. 41, No.2, Feb. 1992, pp. 239-244

[35] DW Davies, DLA Barker "Communication Networks For Computers" Wiley, New York, 1973

[36] W Delaney, E Vaccari "Dynamic Model and Discrete Event Simulation" Manuel Dekker, Inc, USA, 1989

[37] EW Dijkstra "A note on two problems with connection with graphs" Numer. Math Vol.11 Oct 1959 pp. 269-271

[38] J Duato "A New Theory Of Deadlock-Free Adaptive Routing In Wormhole Networks" IEEE Transactions on Parallel and Distribute Systems, Vol. 4, No.12, Dec. 1993, pp. 1320-1331.

[39] J Duato, P López. "Performance Evaluation of Adaptive Routing Algorithms for k-ary n-cubes" K. Bolding and L. Snyder, editors. First International Workshop, PCRCW'94, Vol. 853 of LNCS, May 1994, pp 45-59.

[40] JJ Dujmovic "Universal Benchmark Suites" Proc. Modelling, Analysis and Simulation of Computers and Telecommunicacion Systems. Oct 1999, pp. 197-205

[41] J Duato, S Yalamanchili, L Ni "Interconnection Networks, an Engineering Approach" IEEE Computer Society Press. 1997

[42] R Duncan. "A Survey of Parallel Computer Architectures" IEEE Computers, Vol. 23 No.2 Feb. 1990, pp. 5-16

[43] B Falsafi, DA Wood "Reactive NUMA: A Design for Unifying S-COMA and CC-NUMA" Intl. Symp. On Computer Architecture (ISCA), June 1997 pp.229-240

[44] S Felperin, L Gravano, G Pifarre, J Sanz. "Routing Techniques for Massively Parallel Communication" Proc. of the IEEE, Vol.79, No. 4, April 1991, pp. 488-503

[45] T Feng. "A Survey of Interconnection Networks", IEEE Computer, Vol. 14, No. 12, Dec. 81, pp. 5-20

[46] G Fishman "Conceptos y Métodos en la Simulación Digital de Eventos Discretos" Editorial Limusa, 1a. ed., 1978.

[47] M J Flynn "Some Computer Organisations and their effectiveness" IEEE Transaction on Computers, c-21(9), 1972 pp. 114-118

[48] I Foster "Designing and Building Parallel Programs" Reading: Addison-Wesley, Inc. 1995

[49] D.Franco, I.Garcés, E. Luque "Dynamic Routing Balancing in Parallel Computer Interconnection Networks" Vector and Parallel Processing- Vecpar'98. Selected Papers. Springer Verlag. 1999 pp.921-934

[50] D.Franco, I.Garcés , E. Luque "Balanceo del Encaminamiento Distribuido para la Comunicación en Redes de Interconexión" IV Congreso Argentino en Ciencias de la Computación. CACIC98 Argentina (1998) pp. 665-680

[51] D Franco, I Garcés , E Luque "Distributed Routing Balancing for Interconnection Networks" Procc. of Intl. Conf. on High Perf. Compt. (HiPC98). IEEE Computer Society pp. 253-261

[52] D Franco, I Garcés, E Luque "Avoiding Communication Hot-Spots in Interconnection Networks" Procc. of 32nd Hawaii Intl. Conf. on System Sciences. (HICSS99)

[53] D Franco, I Garcés, E Luque "A new method to make communication latency uniform:". Procc. of ACM International Conference on Supercomputing (ICS99) pp.210-219.

[54] D Franco, I Garcés, E Luque "Analytical Modelling of the Network Traffic Performance" Proc. Modelling, Analysis and Simulation of Computers and Telecommunicacion Systems. IEEE Computer Society . Oct 1999 pp. 190-196

[55] RM Fujimoto "Parallel Discrete Event Simulation" Comm. of the ACM, Vol. 33, No.10, Oct. 1990, pp. 31-53.

[56] M Galles "Scalable Pipelined Interconnect for Distributed Endpoint Routing: The SGI-SPIDER Chip", Proc. Of Hot Interconnects Symp. Aug. 1996

[57] I Garcés "Evaluación del Balanceo Distribuido del Encaminamiento utilizando Caminos Múltiples" Master Thesis. Univ. Autónoma de Barcelona

[58] I.Garcés, D. Franco, E.Luque "Dynamic Routing Balancing: A new technique for Parallel Computer Communication" (Comunicación) VIII Jornadas de Paralelismo. Cáceres. Septiembre 1997 pp.241-250

[59] I. Garcés, D. Franco, E. Luque "Improving Parallel Computer Communication: Dynamic Routing Balancing" Euromicro Workshop on Parallel and Distributed Processing (PDP-98) IEEE CS Press. USA, 1998 pp.111-119

[60] J García, J Duato. "Dynamic Reconfiguration Of Multicomputer Networks: Limitations and Trade-Offs" Euromicro Workshop on Parallel and Distributing Processing, IEEE Computer Society Press, 1993, pp. 317-323.

[61] J García, JL Sánchez, P González "PEPE: A Trace-Driven Simulator to Evaluate Reconfigurable Multicomputer Architectures" Applied Parallel Computing in Industrial Problems and Optimization, J. Wasniewski, J. Dongarra, K. Madsen and D. Olesen (Eds.), Lecture Notes in Computer Science, 1996. p. 302-311

[62] P Gaughan, S Yalamanchili "Adaptive Routing Protocols for Hypercube Interconnection Networks". IEEE Computer, Vol. 26, No. 5, May 1993, pp 12-23

[63] A Geist et al. "PVM. User's Guide and Reference Manual". Oak Ridge National Laboratory. 1994

[64] LR Goke, GJ Lipovski "Banyan networks for partitioning multiprocessing systems" Procc. First Intl. Symp. On Comp. Architecture, 1973, pp.21-28

[65] IS Gopal "Prevention of store-and-forward deadlock in computer networks" IEEE Trans. On Communications, v.Com-33, n.12, Dec 1985 pp.1258-1264

[66] G Gordon. "System Simulation". Prentice-Hall, 2nd. Ed., 1978.

[67] AG Greenberg, B Hajek "Deflection Routing in Hypercube networks" IEEE Trans. On Communications, V.Com.40, n.6, Jun 1992, pp. 1070-1081

[68] JA Gregorio, F Vallejo, R Beivide, C Carrion "Petri Net modelling of interconnection networks for massively parallel architectures" Proc. ACM Int. Conf. On Supercomputing, July 1995

[69] R Gupta, S Pande, K Psarris, V Sarkar "Compilation techniques for parallel systems" Parallel Computing 25(1999) pp.1741-1783

[70] CAR Hoare "Communicating Sequential Processes", Prentice-Hall Int. 1985

[71] RW Hockney, CR Jesshope "Parallel Computers 2" Adam Hilger. 1988

[72] K Hwang "Advanced Computer Architecture: Parallelism, Scalability, Programmability". Mc Graw-Hill, 1993.

[73] K Hwang, FA Briggs "Computer Architecture and Parallel Processing" McGraw-Hill, New York, 1984

[74] Inmos " Occam 2 Toolset User Guide", 1993.

[75] Intel iPSC/1 Reference Manual, Beaverton. OR, 1986

[76] CR Jesshope, JT Yantchev "High performance communications in processor networks" Proc. 16th Symp. Comp. Architecture. 1989. pp. 150-157

[77] F Kamoun, L Kleinrock "Stochastic Performance Evaluation of Hierarchical Routing for Large Networks" Computer Networks Vol.3 Nov 1979 pp. 337-353

[78] P Kermani, L Kleinrock "Virtual Cut-Through: A New Computer Communication Switching Technique". Computer Networks, Vol. 3, 1979, pp. 267-286.

[79] JH Kim, AA Chien "Evaluation of wormhole routed networks under hybrid traffic loads" Proc. Of the Hawaii inter. Conf. On System Sciences, 1993

[80] J Kim, Z Liu, A Chien "Compressionless Routing: A Framework for Adaptive and Fault-Tolerant Routing". Proc. of the 21st Intl. Symposium on Computer Architecture, Apr 1994, pp.289-300

[81] A Law, D Kelton "Simulation Modelling and Analysis". Mc Graw-Hill, 2d. ed, 1991.

[82] CE Leiserson "Fat-trees: Universal networks for hardware-efficient supercomputing" IEEE Trans. On Computers, v.C-34, Oct 1985, pp.892-901

[83] C E Leiserson et al. "The Network Architecture of the Connection Machine CM-5" Journal of Parallel and Distributed Computing 33(2) pp. 145-158

[84] M Livny, J Basney, R Raman, T: Tanenbaum "Mechanisms for High Throughput Computing" SPEEDUP Journal, v.11(1) Jun 1997, pp.36-40

[85] PK Loh, W Jing, C Wentong, N Sriskanthan, "How Network Topology Affects Dynamic Load Balancing" IEEE Parallel & Distributed Technology, Vol. 4, No.3, Fall 1996, pp. 25-35.

[86] E Luque, D Rexachs, J Sorribes, A Ripoll "A modular arbitration system for múltiple buses multiprocessors" Microcomputers, usage and design. K Waldschmidt and B Myhrhaug (eds) Elsevier Science Publishers B.V. (North-Holland) 11th Euromicro Symp. On microprocessing and microprogramming, 1985 pp.579-585

[87] MH MacDougall "Simulating Computer Systems Techniques and Tools". The MIT Press, 1987.

[88] TG. Mattson, G Henry "The ASCI Option Red Supercomputer" Intel Corporation. Disponible en http://www.cs.sandia.gov/ISUG97/papers/Mattson/OVERVIEW.html

[89] M May, P Thompson Eds. "Networks, Routers and Transputers: Function, Performance and Applications" IOS press, 1993.

[90] PK McKinley, Y-J Tsai, D Robinson "Collective Communication in Wormhole-Routed Massively Parallel Computers". IEEE Computer, Vol. 28, No. 12, Dec. 1995 pp.39-50

[91] PK McKinley, C Trefftz "Multisim: A Tool for the Study of Large Scale Multiprocessors" Proc. of the 1993 International Workshop on Modelling, Analysis, and Simulation of Computer and Telecommunication Networks (MASCOTS), San Diego, California, Jan. 1993,pp. 57-62.

[92] J Miguel, A Arruabarrena, R Beivide "Conservative Parallel Discrete Event Simulation in a Transputer Based Multicomputer". In "Transputer Applications and Systems '93" R. Grebe et al (eds.) IOS-Press 1993, pp 636-650.

[93] J Miguel, A Arruabarrena, R Beivide. "Simulación de Sistemas de Sucesos Discretos en Arquitecturas Masivamente Paralelas". IV Reunión de Paralelismo, 1993.

[94] J Miguel, A Arruabarrena, C Izu, R Beivide. "Simulación Paralela de una red de Encaminamiento de Mensajes" V Jornadas de Paralelismo, 1994, pp. 62-73

[95] D Min , MW Mutka "Determining External Contention Delay due to Job Interactions in a 2D Mesh Wormhole Routed Multicomputer" Proc. IEEE Symp. Parallel and Distributed Processing Dic. 1993 pp.258-265

[96] D Min, MW Mutka "A Model For Analysing Interactions In 2D Mesh Wormhole-Routed Multicomputers" Parallel Computing, No. 22, 1996, pp. 675-699.

[97] J Misra. "Distributed Discrete-Event Simulation". Computer Surveys, Vol. 18, No.1, March 1989, pp. 39-65.

[98] Message Passing Interface Forum "MPI. A Message-Passing Interface Standard". International Journal of Supercomputer Application and High Performance Computing, v.9, no.3/4. 1994

[99] MW Mutka, PK McKinley "Supporting a Simulation Environment with OpenSim" Simulation, Vol. 61, No. 4, Oct. 1993, pp. 223-235.

[100] L Ni, C Glass "The Turn model for Adaptive Routing". Proc. of the 19th International Symposium on Computer Architecture, IEEE Computer Society, May 1992, pp. 278-287

[101] MG Norman, P Thanisch "Models of machines and computation for mapping in multicomputers" ACM Comp. Surveys, Vol.25, N.3, Sep 1993, pp.263-302

[102] D Nussbaum, A Agarwal "Scalability of Parallel Machines". Communications of the ACM. March 1991, Vol. 34, No. 3, pp 57-61

[103] JH Patel "Performance of Processor-memory interconnections for multiprocessors" IEEE Trans. On Computers, v. C-30 Oct 1981 pp.771-780

[104] PCRCW'94. http://www.cs.washington.edu/ research/ projects/ lis/ chaos /www/ presentation.html

[105] MJ Pertel "A Simple Simulator for Multicomputer Routing Networks". CalTech-CS-TR-92-04, 1992

[106] F Petrini, M Vanneschi "SMART: A Simulator of Massive Architectures and Topologies" Proc. of the Parallel and Distributed Systems Euro-PDS'97, Jun. 1997, pp 185-191

[107] GP Pfister, A Norton. "Hot-Spot Contention and Combining in Multistage Interconnection Networks" IEEE Transactions on Software Engineering, Vol. C-34, No. 10, Oct. 1985, pp. 943-948

[108]M De Prycker "Asynchronous Transfer Mode, solutions for Broadband ISDN" Prentice-Hall 1995 (3d Ed)"

[109] V Puente "Impacto del subsistema de comunicación en el rendimiento del los computadores paralelos: desde el Hardware hasta las aplicaciones" PhD Thesis. Univ. Of Cantabria. Oct. 1999

[110] GJ McRae "How application domains define requirements for the grid" Communications of the ACM, 40 (11): Nov 1997 pp.75-84

[111] DA Reed, DC Grunwald "The Performance of Multicomputer Interconnection Networks", IEEE Computer, Vol. 20, No. 6, April 1987, pp. 63-73

[112] DA Reed, AD Malony, BD McCredie "Parallel Discrete Event Simulation using Shared Memory" IEEE Transactions on Software Engineering, Vol. 14, No. 4, April 1988, pp. 541-553

[113] J Rexford, W Feng, J Dolter, K Shin "PP-MESS-SIM: A Flexible and Extensible Simulator for Evaluating Multicomputers Networks". IEEE Transactions on Software Engineering, Vol. 8, No. 1, Jan 1997, pp. 25-39

[114] AW Roscoe "Routing messages through networks: an exercise in deadlock avoidance" Tech. Report, Oxford University Computing Laboratory Report, 1987

[115] SM Ross "Stochastic Processes" Wiley, New York, 1983

[116] L Schwiebert, DN Jayasimha "A Universal Proof Technique for Deadlock-free Routing in Interconnection Networks "Symp. On Parallel Algorithms and Architectures Jul 1995, pp.175-184

[117] SL Scott "Synchronization and Communication in the T3E Multiprocessor" Procc. Of ASPLOS VII, Oct 1996 pp. 26-36

[118] CL Seitz "The Cosmic Cube" Communications of the ACM Vol.28 No 1Jan 1985 pp. 22-33

[119] CL Seitz "Multicomputers. Developments of Concurrency and Communication" Addison-Wesley, 1990, pp.131-200

[120] J Siegel et al "Using the multistage cube network topology in parallel supercomputers" Proc. Of the IEEE, V.77, Dec. 1989, pp. 1932-1953

[121] J Siegel "Interconnection Networks for Large-Scale Parallel Processing. Mc Graw-Hill, 2d. Edition, 1990

[122] D Sima, T Fountain, P Kacsuk "Advanced Computer Architectures. A design space approach" Adisson-Wesley 1997

[123] M Snir, P Hochschild, DD Frye, KJ Gildea "The communication software and parallel environment of the IBM SP2" IBM Systems Journal. Vol.34, N.2, pp. 205-221.

[124] T Sterling, D Becker, DF Savarese et al. "BEOWULF: A Parallel Workstation for Scientific Computation" Procc. Of the Intl. Conf. On Parallel Processing, 1995 pp. 11-14

[125] CB Stunkel et al. "The SP2 high-performance switch" IBM Systems Journal, v.34, no2 Aug. 1994, pp.185-204

[126] R Suaya, G Birtwistle. "VLSI And Parallel Computation Frontiers". Morgan Kaufmann Publishers, 1990.

[127] VS Sunderam, GA Geist "Heterogeneous parallel and distributed computing" Parallel Computing 25 (1999) pp. 1699-1721

[128] AS Tanenbaum "Computer Networks" Prentice-Hall. Englewood Cliffs, NJ, USA. 3rd Ed. 1997

[129] T Thiel "The Connection Machines CM-1 and CM-2" http://mission.base.com/tamiko/cm/, March 1994

[130] LG Valiant, GJ Brebner "Universal Schemes for Parallel Communication" ACM STOC. Milwaukee 1981, pp 263-277

[131] LG Valiant "A scheme for fast parallel communication" SIAM J. Comp. V.11 No.2, May 1982 pp. 350-361

[132] LG Valiant "A bridging model for parallel computation" Communications of the ACM, 33(8), Aug 1990 pp.103-111

[133] PH Welch "Parallel Hardware and Parallel Software: A Reconciliation" Internal Report. Comp. Laboratory. Univer. Of Kent at Canterbury, England

[134] M Wilkes, A Hopper "The collapsed LAN: a Solution to a Bandwidth Problem?" Computer Architecture News, Jun 1997 pp. 1-5

[135] B Wilkinson. "Computer Architecture. Design and Performance", 2d. Edition, Prentice-Hall, 1996.

[136] CL Wu, TY Feng "On a class of multistage interconnection networks" IEEE Trans. on Computers, v.C-29 Aug 1980 pp.694-702

[137] www.nhse.org/grand_challenge.html

[138] J Yantchev, C Jesshope. "Adaptive, Low Latency, Deadlock-free Packet Routing Networks of Processors". IEEE Proceedings, Vol. 136, Pt. E, No.3, May 1989, pp 178-186.

[139] B Zeigler "Multifaceted Modelling and Discrete Event Simulation". Cap 1. Academic Press Inc., 1984.

Apéndice A
Análisis del modelo de contención de dos canales

A. 1 Introducción

En este apéndice se describe el análisis del retardo de contención para un modelo de dos canales que colisionan en un único punto. Este análisis concluye con la deducción de la fórmula Min-Mutka usada en el resto del capítulo. Esta explicación está resumida de la que se encuentra en [95] y se presenta aquí por completitud.

Supóngase que tenemos un sistema en el que colisionan dos canales en un único punto como el mostrado en la Figura 3A 1. Asumimos que es el único punto de colisión para ambos canales en la red de interconexión.

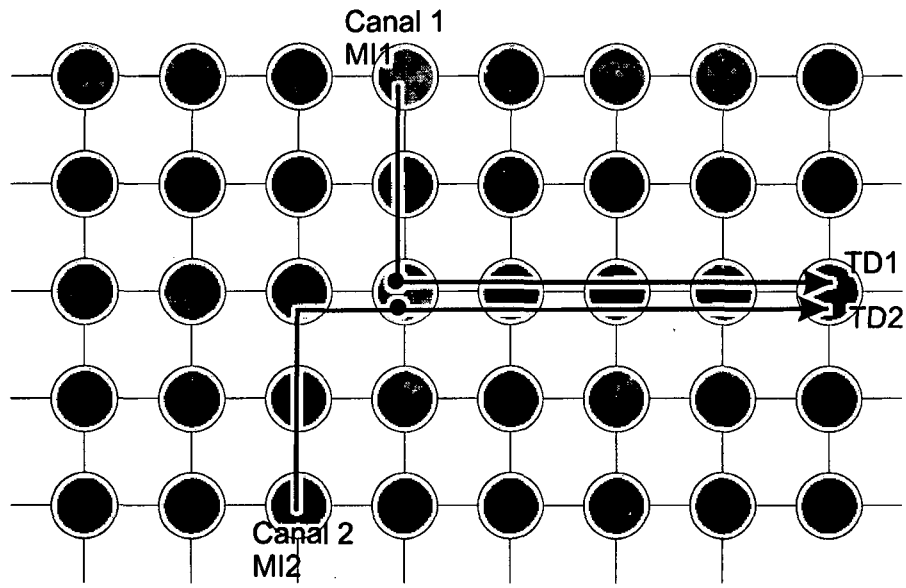


Figura 3A 1 Sistema de dos canales

Los canales están caracterizados por los parámetros de intervalo entre inyección de mensajes y tiempo de transmisión; MI1, TD1 para el canal 1 y MI2, TD2 para el canal 2. Después del punto de colisión, ambos canales presentan un tiempo de ocupación del canal o tiempo de transmisión que es conocido y constante, a los que hemos llamado TD1 y TD2, respectivamente.

Sean T1 y T2 las variables aleatorias que representan los tiempos locales de cómputo entre sucesivos mensajes en los nodos fuente de ambos canales cuyas medias son, respectivamente, MI1 y MI2. Asumimos que T1 y T2 son variables aleatorias independientes, pues representan dos tareas en ejecución y no tienen ninguna relación entre ellas. Sea, asimismo, Ck, donde k puede ser 1 ó 2, la variable aleatoria del retardo debido a la contención sobre el canal k cuando colisionan en canal 1 y el canal 2. Entonces, el retardo de contención sobre el canal k es Ck + TDk.

Si ambos MI1 y MI2 fuesen iguales, el sistema se reduciría a un modelo de comunicaciones de colas y podría usarse la teoría de colas para calcular el retardo de espera en la cola. Sin embargo, esta aproximación no contempla el caso de que MI1 y MI2 sean diferentes y no es capaz de proveer un resultado preciso. En este caso debemos utilizar una aproximación estocástica para desarrollar una fórmula de predicción del retardo de contención Ck sobre el canal k. Lo que buscaremos calcular, pues, será la esperanza de Ck, E[Ck].

La esperanza de Ck puede ser calculada sumando las esperanzas condicionales de Ck, para todas las posibles condiciones que pueden darse sobre las relaciones entre (T1 y T2) y (TD1 y TD2). Todos los posibles casos son los siguientes:

Caso 1: T1 < TD2 y T2 < TD2

Caso 2: T1 < TD2 y T2 >= TD2

Caso 3: T1 >= TD2 y T2 < TD2

Caso 4: T1 >= TD2 y T2 >= TD2

La esperanza sobre el retardo de contención Ck sobre el canal k es la suma de la esperanza del retardo de contención condicionada a que se dé cada uno de los casos ponderada por las probabilidades de aparición de cada caso, tal y como expresa la Ecuación 3A.1:

$$E[Ck]=\sum_{n=1}^{4}\Pr ob\{caso_n\}*E[C_k \mid caso_n]$$

Ecuación 3A.1

Los siguientes puntos presentan cómo calcular la probabilidad de aparición y la esperanza condicionada del retardo de contención para cada caso.

A. 2 Cálculo de E[Ck]. Caso 1

La probabilidad de aparición del caso 1 se calcula del siguiente modo:

$$\Pr ob\{caso_1\}=\Pr ob\{T1<TD2 \text{ y } T2<TD1\}\overset{(1)}{=}\Pr ob\{T1<TD2\}*\Pr ob\{T2<TD1\}$$

(1)Dado que T1 t T2 son independientes

Para calcular la esperanza condicionada del retardo de contención de T1 debido a T2 en el caso 1, debemos fijarnos en un ejemplo de la situación del caso 1 mostrado en la Figura 3A 2. Puesto que T1 y T2 son menores que TD2 y TD1, respectivamente, esta situación se repetirá para el periodo de solapamiento entero entre T1 y T2 aunque T1 y T2 sean variables aleatorias. Por lo tanto,

$$E[C1 \mid caso_1]=TD2-E[T1 \mid T1<TD2]$$

La razón de usar la esperanza condicionada $E[T1 \mid T1<TD2]$ en lugar de $E[T1]$ simplemente es que T1<TD2 es la condición dada del caso 1. El cálculo de $E[T1 \mid T1<TD2]$ se realiza mediante el cálculo de una probabilidad condicionada. Al final del apéndice se muestra un ejemplo, suponiendo que T1 tiene una distribución exponencial con media MI1. La $E[C2 \mid caso_1]$ puede calcularse de una forma similar.

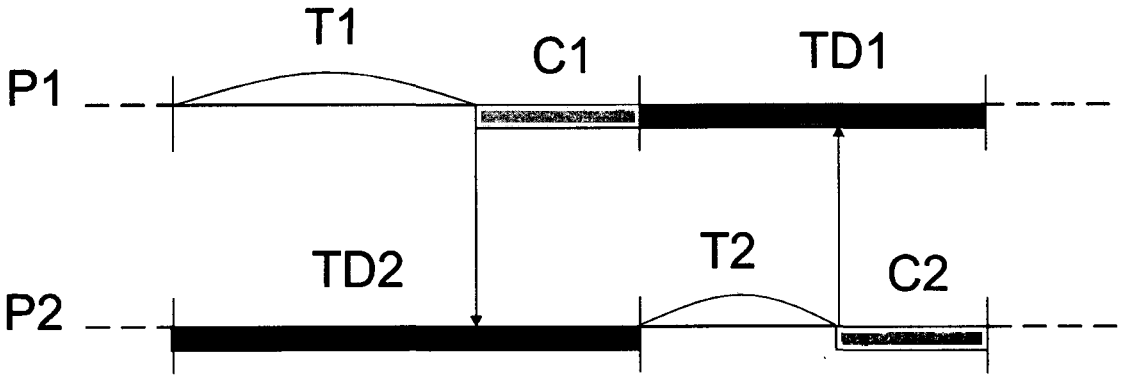


Figura 3A 2 Retardo de contención de 2 canales. Caso 1

A. 3 Cálculo de E[Ck]. Caso 2

La probabilidad del caso 2 se puede calcular similarmente como en el caso 1, es decir, $\Pr ob\{caso_2\} = \Pr ob\{T1 < TD2\} * \Pr ob\{T2 >= TD1\}$. El cálculo de la esperanza condicionada del retardo de contención bajo el caso 2 es complicado. Debemos considerar dos posibles temporizaciones de la llegada del mensaje de P2 dependiendo del estado de P1 cuando el mensaje de P2 llega al punto de colisión.

Una de las posibles situaciones, que se muestra en la Figura 3A 3 (a), es que el mensaje de P2 llegue al punto de colisión mientras el enlace esta libre. En este caso, el mensaje puede ser transmitido sin ningún retardo, pero causa un retardo de contención al mensaje que llega al punto de colisión desde P1 durante la transmisión del mensaje de P2. Este retardo de contención, llamado C1a, es TD2 menos el tiempo restante de T1 después de que el mensaje de P2 llega al punto de colisión. La tarea de calcular C1a es calcular este tiempo restante. Por medio de la teoría de renovación de los procesos estocásticos desarrollamos una aproximación.

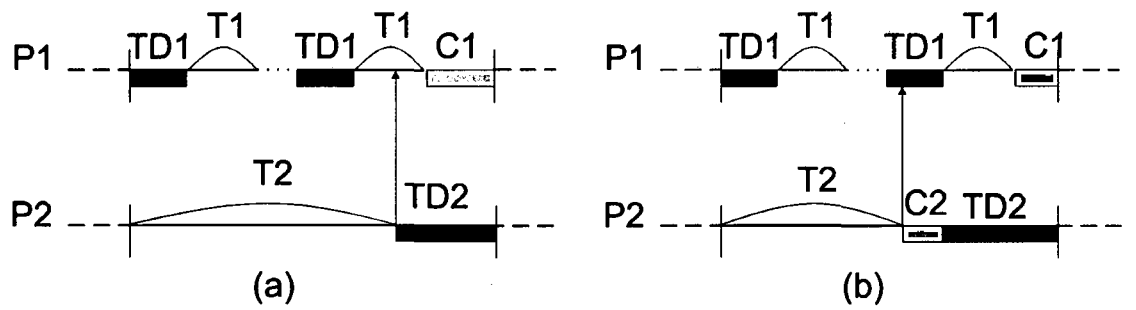


Figura 3A 3 Retardo de contención de 2 canales. Caso 2

Supongamos un proceso cuya media del tiempo entre llegadas es μ y cuya varianza es σ^2. La esperanza del tiempo restante R_t desde un tiempo dado t hasta la llegada del próximo evento generado por el proceso es $E[R_t] = \frac{1}{2}\mu(1 + \mu^2/\sigma^2)$. Una

demostración de esta ecuación se da al final de este apéndice. Nótese que el tiempo restante es aproximadamente mayor que un medio del tiempo medio entre llegadas.

Por lo tanto, el tiempo restante aproximado de T1 puede calcularse como $\frac{1}{2}E[T1 \mid T1 < TD2]$, y así, C1a es $TD2 - \frac{1}{2}E[T1 \mid T1 < TD2]$. Puesto que T2 es mayor o igual que TD1, C1a es el retardo de transmisión después de varias transmisiones. Así, la esperanza del retardo de contención para un mensaje, E[C1]a, es C1a/E[N_t], donde E[N_t] es el número medio de transmisiones. Aproximadamente,

$$E[N_t] = \left\lceil \frac{TD2 + E[T2 \mid T2 >= TD1]}{TD1 + E[T1 \mid T1 >= TD2]} \right\rceil$$

donde N_t es la variable aleatoria del número de transmisiones y $\lceil\ \rceil$ indica el entero mayor más cercano. Resumiendo, la esperanza del retardo de contención para un mensaje de P1 y P2 en el caso de la parte a se dan en las siguientes ecuaciones:

$$E[C1]a = C1a / E[N_t], \qquad \text{E[C2]}a = 0$$

La otra posible temporización de las llegadas de los mensajes de P2 es que el mensaje de P2 llegue al punto de colisión cuando el enlace está siendo usado por un mensaje de P1. Esta situación se muestra en la Figura 3A 3 (b). En este caso, ambos caminos tienen retardo de contención. C1b y C2b representan los retardos de contención de P1 y P2, respectivamente. C1b se puede calcular de manera similar como si fuera un caso 1, es decir, $C1b = TD2 - E[T1 \mid T1 < TD2]$. Como en la parte (a), C1b es el retardo de contención después de varias transmisiones. La esperanza el retardo de contención por mensaje de P1 es C1b dividido el número medio de transmisiones, E[N_t]. Es decir,

$$E[C1]b = (TD2 - E[T1 \mid T1 < TD2]) / E[N_t]$$

El retardo de contención de P2, C2b, es el tiempo de comunicación restante de P2 después de la llegada del próximo mensaje. Como una aproximación, es un medio de TD1. Puesto que C2b puede ocurrir en cada transmisión, $E[C2]b = \frac{1}{2}TD1$.

Sea la probabilidad del caso (a) Pr(a), y la probabilidad del caso (b) Pr(b). La esperanza del retardo de contención de P1 y P2 en el caso que T1<TD2 y T2>=TD1 se calcula como se indica a continuación:

$$E[C1 \mid caso_2] = \Pr(a)E[C1]a + \Pr(b)E[C1]b$$
$$E[C2 \mid caso_2] = \Pr(a)E[C2]a + \Pr(b)E[C2]b$$

Pr(a) es el porcentaje que el enlace del punto de colisión es reservado por P2 sin ningún retardo de contención para varias transmisiones de P1. Esta probabilidad puede aproximarse por medio de la teoría de la renovación como:

$$\Pr(a) = E[T1 \mid T1 < TD2] \Big/ (TD1 + E[T1 \mid T1 < TD2])$$

Y como Pr(b) es 1-Pr(a), entonces

$$\Pr(b) = 1 - \Pr(a) = TD1 \Big/ (TD1 + E[T1 \mid T1 < TD2])$$

La demostración de la ecuación para Pr(a) es una aplicación del teorema básico de la renovación. En el caso de que el canal se considere un sistema que alterna entre dos estados, se da una demostración en [115].

A. 4 Cálculo de E[Ck]. Caso 3

El caso 3 es exactamente el mismo al caso 2 si todas las notaciones sobre P1 se cambian por las de P2 y todas las de P2 por las de P1. Por lo tanto, omitimos la explicación para este caso.

A. 5 Cálculo de E[Ck]. Caso 4

La probabilidad del caso 4 es

$$\Pr ob\{caso_4\} = \Pr ob\{T1 >= TD2\} * \Pr ob\{T2 >= TD1\}.$$

Debido a que T1 y T2 son mayores o iguales que TD2 yTD1, respectivamente, ambas E[C1| caso$_4$] y E[C2| caso$_4$] se calcula usando la teoría de la renovación como se hizo en el caso 2. La Figura 3A 4 ilustra como calcular E[C1| caso$_4$]. Como antes, consideramos dos posibles situaciones dependiendo de la temporización relativa de los mensajes de ambos canales. Si un mensaje de P2 llega durante T1 como se muestra en la parte (a) de la figura mencionada, entonces el tiempo de espera de este mensaje es cero. Si un mensaje de P2 llega durante TD1, entonces el tiempo de espera es aproximadamente un medio de TD1. De acuerdo con la teoría de la renovación, la probabilidad de la parte (a), Prob(a), es $\Pr(a) = E[T1 \mid T1 < TD2] \Big/ (TD1 + E[T1 \mid T1 < TD2])$ y la probabilidad de la parte (b), Prob(b), es 1 –Prob(a). Por consiguiente, el retardo de contención de P1 es

$$E[C1 \mid caso_4] = \Pr(b) * TD1 / 2$$

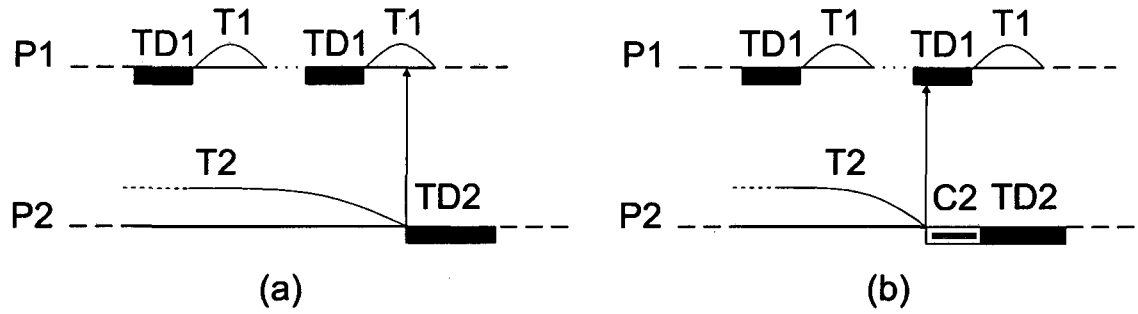


Figura 3A 4 Retardo de contención de 2 canales. Caso 4

Similarmente, $E[T2 \mid caso_4]$ puede calcularse como:

$$E[C2 \mid caso_4] = \frac{TD1}{TD1 + E[T1 \mid T1 >= TD2]} * \frac{TD2}{2}$$

A. 6 Formula de Min-Mutka

Con este último caso finaliza la deducción de la ecuación de cálculo del retardo en un sistema de dos canales con un único punto de colisión. La Ecuación 3A.1 es la fórmula Min-Mutka usada en el capítulo 3 cuando se le llama con los parámetros de descripción de los canales. Esta fórmula devuelve E[C2]+TD1 si se le invoca como Min-Mutka (M1,TD1,M2,TD2,1) y E[C2]+TD2 si se le invoca como Min-Mutka (M1,TD1,M2,TD2,2).

A. 7 Cálculo de la probabilidad condicionada para una distribución exponencial

En este punto, vamos a suponer que las variables aleatorias que representan la llegada de mensajes en los canales 1 y 2 tienen una función de distribución estadística exponencial. Este tipo de distribución es la que responde a un comportamiento estocástico de un proceso donde los sucesos son independientes uno de otro, de manera que la llegada de un suceso no condiciona la llegada de ningún otro, pasado o futuro.

Por esta razón estos sistemas se suelen denominar "sin memoria" y son una buena aproximación para representar la cadencia de generación de mensajes por una tarea que esta computando y de la cual no se tiene ninguna información sobre la naturaleza de los cálculos que realiza. Esta tarea se comporta de manera que, después de una porción de cómputo, realiza una acción de comunicación y, a continuación, empieza otra porción de cómputo que no tiene relación con la anterior, con lo que su duración es independiente de la duración de la computación anterior.

A continuación, se muestran los cálculos de las probabilidades condicionadas E[X|X<k] y E[X|X>=k], para una distribución exponencial de X. La distribución exponencial tiene una función de densidad $f_x(X)=\lambda e^{-\lambda x}, x>0$ y una función de distribución, integral de la anterior, $F_x(X)=1-e^{-\lambda x}, x>0$. Esta distribución tiene como media $\mu = 1/\lambda$ y como varianza $\sigma^2 = 1/\lambda^2$.

A.7.1. Esperanza condicionada E[X|X<k]

Para calcula la esperanza condicionada E[X|X<*k*] debemos hallar la probabilidad condicionada de que se dé el caso de que la variable X sea menor que *k*. Esto equivale a hallar el valor de la función de distribución bajo esa condición:

$$F_x(X \mid X<k)=\Pr ob\{X<=x \mid X<k\}=\frac{\Pr ob\{X<=x \cap X<k\}}{\Pr ob\{X<k\}}=$$

$$\begin{cases} 1, & (x>k) \\ \dfrac{\Pr ob\{X<=x\}}{\Pr ob\{X<k\}}=\dfrac{1-e^{-\lambda x}}{1-e^{-\lambda k}}, & (x<=k) \end{cases}$$

A continuación, se calcula la función de densidad de probabilidad f[X|X<=*k*] derivando la expresión anterior,

$$f_x(X \mid X<=k)(x)=\frac{1}{1-e^{-\lambda k}}\lambda e^{-\lambda x}$$

Finalmente, la esperanza condicionada buscada es la integral desde 0 hasta *k* de la función de densidad de probabilidad multiplicada por x:

$$E[X \mid X<=k]=\frac{\lambda}{1-e^{-\lambda k}}\int_0^k x\lambda e^{-\lambda x}dx=$$

$$=\frac{\lambda}{1-e^{-\lambda k}}\left[\frac{e^{-\lambda x}}{-\lambda}\left(x+1/\lambda\right)\right]_0^k=\frac{\lambda}{1-e^{-\lambda k}}\left[\frac{e^{-\lambda k}}{-\lambda}\left(k+1/\lambda\right)-\frac{1}{-\lambda}\frac{1}{\lambda}\right]=$$

$$=\frac{1/\lambda-\left(k+1/\lambda\right)e^{-\lambda k}}{1-e^{-\lambda k}}$$

A.7.2. Esperanza condicionada E[X|X>=k]

Similarmente, calculamos ahora la esperanza condicionada de X bajo el supuesto de que X>=*k*. Hallamos la función de distribución condicionada:

$$F_x(X \mid X >= k) = \Pr ob\{X <= x \mid X >= k\} = 1 - \Pr ob\{X > x \mid X >= k\} =$$

$$= 1 - \frac{\Pr ob\{X > x \cap X >= k\}}{\Pr ob\{X >= k\}} =$$

$$\begin{cases} 1 - \dfrac{\Pr ob\{X >= k\}}{\Pr ob\{X >= k\}} = 0, & (x < k) \\ 1 - \dfrac{\Pr ob\{X >= x\}}{\Pr ob\{X >= k\}} = \dfrac{1 - e^{-\lambda x}}{1 - e^{-\lambda k}} = 1 - \dfrac{\int_x^\infty \lambda e^{-\lambda t} dt}{\int_k^\infty \lambda e^{-\lambda t} dt} = 1 - \dfrac{e^{-\lambda x}}{e^{-\lambda k}} = 1 - e^{-\lambda(x-k)}, & (x >= k) \end{cases}$$

A continuación, calculamos la función de densidad derivando la función de distribución:

$$f_x(X \mid X > k)(x) = \lambda e^{-\lambda(x-k)}$$

Y, finalmente, integramos la función de densidad multiplicada por x en el intervalo desde *k* hasta infinito para obtener la esperanza condicionada:

$$E[X \mid X >= k] = \int_k^\infty x \lambda e^{-\lambda(x-k)} dx = \lambda e^{\lambda k} \int_k^\infty x e^{-\lambda x} dx =$$

$$= \lambda e^{\lambda k} \left[\frac{e^{-\lambda x}}{-\lambda} \left(x + 1/\lambda \right) \right]_k^\infty = k + 1/\lambda$$

A. 8 Esperanza del tiempo restante, Rt

Considérese un proceso de renovación cuyos intervalos de llegada de sucesos $T_0, T_1, \ldots$son variables aleatorias con media μ y varianza σ. Aunque no necesitamos especificar la distribución de los T_i, por conveniencia, denotamos F(x) y f(x) las funciones de distribución de probabilidades y de densidad de probabilidad, respectivamente. Sea R_t el tiempo entre la última renovación y la siguiente renovación, y G(x) y g(x) sean las funciones de distribución de probabilidades y de densidad de probabilidad de R_t, respectivamente.

De la teoría de renovación [115], sabemos que

$$\lim_{t \to \infty} \Pr ob\{R_t < x\} = \frac{1}{\mu} \int_0^x (1 - F(y)) dy$$

Y, por lo tanto, $g(x) = (1 - F(x))/\mu$. Usando este hecho, podemos calcular las esperanza de R_t como

$$E[R_t] = \int_0^\infty xg(x)dx = \frac{1}{\mu}\int_0^\infty x(1-F(x))dx = \frac{1}{2\mu}\int_0^\infty x^2 f(x)dx =$$

$$= \frac{1}{2\mu}(\sigma^2 + \mu^2) = \frac{\mu}{2}\left(1 + \frac{\sigma^2}{\mu^2}\right)$$