Implementación de una red reconfigurable en un sistema multi-nivel basado en transputers *

J.L. Sánchez
Dpto. Informática
Escuela Politécnica
Univ. Castilla-La Mancha
Albacete, SPAIN 02071
Fax: 34 67 599224
jsanchez@info-ab.uclm.es

F.J. Alfaro
Dpto. Informática
Escuela Politécnica
Univ. Castilla-La Mancha
Albacete, SPAIN 02071
Fax: 34 67 599224
fjalfaro@info-ab.uclm.es

J.M. García
Dpto. Informática y Sistemas
Facultad Informática
Univ. Murcia
Murcia, SPAIN 30080
Fax: 34 68 364151
jmgarcia@dif.um.es

Resumen

El principal problema en redes de interconexión con encaminamineto segmentado está relacionado con la congestión debida al bloqueo de los mensajes. Las redes reconfigurables son una alternativa para reducir el efecto negativo que dicha congestión produce en el rendimiento de la red. La reconfiguración puede realizarse de diversas formas. Nuestro trabajo está enfocado hacia las redes reconfigurables dinámicamente. Esta técnica consiste básicamente en situar los diferentes nodos en aquellas posiciones en la red que, en cada momento de la computación y de acuerdo al patrón de comunicaciones que hay entre ellos, son las más adecuadas para el desarrollo de dicha computación.

Nosotros estamos aplicando esta técnica a sistemas de transputers reconfigurables. El interés por este tipo de sistemas ha crecido en los últimos años y actualmente hay varios en estado de experimentación. En este trabajo mostramos en qué consisten, indicamos sus características más importantes y apuntamos una posible realización.

1 Introducción

En máquinas masivamente paralelas, un conjunto de nodos trabajan para resolver cierto tipo de problemas. Los nodos comunican datos y coordinan esfuerzos mediante el envío y recepción de mensajes a través de una red de interconexión que los une. En consecuencia, el rendimiento de estas máquinas depende de una manera crítica del rendimiento de sus redes de interconexión. El rendimiento de la red depende de una amplia variedad de factores, no sólo de tipo arquitectónico (topología, encaminamiento, control de flujo, etc), sino también relacionados con las características de la comunicación que se mantenga en cada caso (modelo de tráfico, tamaño de los mensajes, etc). Es importante, pues, incrementar el rendimiento de la red de interconexión, para lo cual el diseñador maneja varios de los aspectos antes citados.

Muchas de las redes actuales usan las técnicas de control de flujo denominadas virtual cut-through y wormhole [4, 7], con las que se reduce la latencia de los mensajes. Cada mensaje es dividido en un determinado número de flits (unidades de control de flujo),

*Este trabajo ha sido financiado en parte por la CICYT por medio de la ayuda TIC94-0510-C02-02

gobernando la ruta el flit de cabecera. Según avanza la cabecera a lo largo de la ruta especificada, el resto de flits le siguen en una manera segmentada. Las características de esta técnica hacen que la latencia de los mensajes sea prácticamente insensible a la distancia que éstos recorren.

El principal problema en redes wormhole son las situaciones de bloqueo. Una situación de bloqueo aparece cuando el flit cabecera, a la llegada a un nodo intermedio, no encuentra un canal libre por el que continuar, y es bloqueado en dicho nodo. El resto de flits van quedando en los tampones de los demás nodos por los que ha pasado la cabecera. Cuando son muchos los mensajes implicados, mayor es la congestión en la red. En este tipo de redes, los efectos de estos acoplamientos se reflejan en el rendimiento de la red, muy sensible a este tipo de situaciones. La congestión, por tanto, puede reducir considerablemente el rendimiento global de la red.

Se han propuesto varias técnicas para reducir o evitar la congestión: canales virtuales, encaminamiento aleatorio o combinación de mensajes. Con canales virtuales se reduce la congestión sólo si el número de conflictos es bajo. Además, su implementación resulta costosa [2], a la vez que se incrementa considerablemente el retraso en cada nodo. La principal desventaja del encaminamiento aleatorio es que no explota la localidad [1]. Por otra parte, el encaminamiento adaptativo es más complejo que el determinista, y reduce la frecuencia de reloj.

En este trabajo, presentamos otra técnica igualmente orientada a mejorar el rendimiento de la red de interconexión minimizando la congestión originada por los mensajes que por ella circulan. Esta técnica está basada en la capacidad de reconfiguración de la red de interconexión, y consiste básicamente en situar los diferentes nodos en aquellas posiciones en la red que resulten más adecuadas para el desarrollo de la computación. Esto se puede conseguir estática o dinámicamente. En el primer caso se hace uso de topologías que permanecen fijas a lo largo de toda la computación, mientras que en el segundo se permite cambiar dichas topologías arbitrariamente a lo largo de la ejecución. Nos ocupamos aquí del segundo grupo.

En este artículo mostramos una red reconfigurable para un multicomputador donde cada nodo consiste en un transputer T9000 y un encaminador C104. La capacidad de cambio se basa en el algoritmo de reconfiguración que hemos diseñado, y que decide cuando se deben llevar a cabo las alteraciones en las posiciones de los nodos. Hemos considerado la reconfiguración de forma limitada, permitiendo que los nodos intercambien su posición con sus vecinos y conservando la topología original. De esta forma no es necesario modificar los algoritmos de encaminamiento, tan sólo las tablas donde se relaciona cada nodo con la posición que ocupa en cada momento, por lo cual cuando se produce un cambio entre dos nodos, debe informarse de dicho cambio al resto. Nuestro algoritmo se ejecuta en cada nodo cuando llega un nuevo mensaje y evalúa el nivel de congestión que se está produciendo, decidiendo si un cambio de posición puede ser favorable. Dicho algoritmo se basa en una función de coste y sólo requiere información local.

La organización del resto del artículo es la siguiente: en el punto 2 se hace una breve introducción a los multicomputadores reconfigurables y presentamos nuestro algoritmo para la reconfiguración dinámica de la red de interconexión. En el punto 3 mostramos varios aspectos relativos a la implementación, y por último, en el punto 4 incluimos las conclusiones finales.

2 Redes reconfigurables

La mayoría de los sistemas basados en el paso de mensajes están implementados mediante topologías fijas. En arquitecturas especializadas, la topología es seleccionada de tal forma que contempla los requerimientos de comunicación de una aplicación determinada. Para arquitecturas de propósito general, se diseñan mecanismos de encaminamiento que permitan a un nodo comunicar con cualquier otro que no tenga como vecino.

El propósito de las redes reconfigurables es reducir el coste de las comunicaciones. La técnica utilizada consiste básicamente en situar los diferentes nodos en aquellas posiciones en la red que, en cada momento de la computación y de acuerdo al patrón de comunicaciones que hay entre ellos, son las más adecuadas para el desarrollo de tal computación.

Una red reconfigurable tiene las siguientes ventajas [5]:

- La programación de una aplicación paralela resulta más independiente de la arquitectura, y es ésta la que finalmente acaba adaptándose a aquella.

- Es fácil explotar la localidad en las comunicaciones.

- Reduce la congestión debido al bloqueo de los mensajes. Este problema es más importante en redes con encaminamiento determinista.

- Resulta muy adecuada para aplicaciones paralelas cuyo patrón de comunicaciones no es conocido a priori e incluso varía con el tiempo.

Básicamente hay dos tipos de reconfiguración: estática y dinámica. En este trabajo, nos centramos en el segundo caso, es decir, la situación de los nodos puede cambiar casi arbitrariamente en tiempo de ejecución.

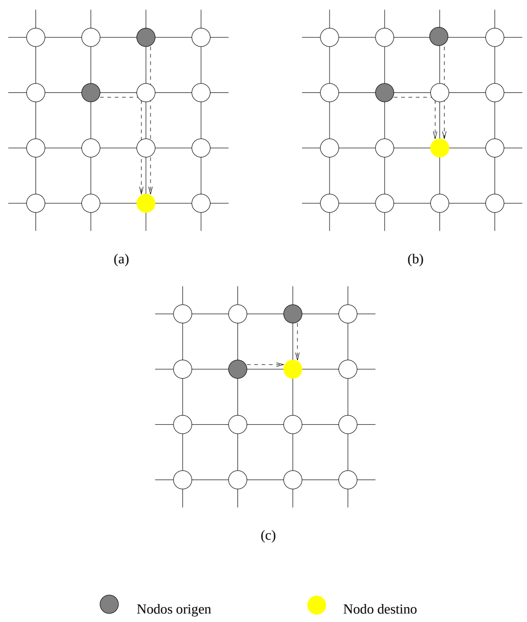

Figura 1: Ejemplo de los efectos del algoritmo de reconfiguración

En la figura 1 se muestran los efectos que puede tener esta técnica para una situación muy elemental. En (a) y (b) los mensajes enviados por los nodos deben atravesar, en algunas ocasiones, los mismos canales hasta alcanzar su destino, originando retrasos en la comunicación debido a las situaciones de bloqueo. Una red reconfigurable tiene la posibilidad de cambiar la posición de los nodos en la red, y de esta forma reducir la congestión. En el ejemplo de la figura, el nodo destino es desplazado hasta la posición indicada en (c). Una vez que se llega a dicha situación, la congestión desaparece, acelerando así el envío y recepción de los mensajes y aumentando el rendimiento de la red.

El nodo que recibe los mensajes se coloca finalmente en su mejor posición en la red. Se ha llevado a cabo mediante pequeñas alteraciones, cambiando su posición con algún nodo vecino.

2.1 Algoritmo para la reconfiguración

Una red reconfigurable está controlada por un algoritmo de reconfiguración. Se presenta a continuación una versión actualizada del algoritmo inicialmente propuesto [6]. Esta versión inicial fue diseñada para trabajar en redes con un control de flujo almacenamiento-y-reenvío (store-and-forward). Hoy en día, la mayoría de multicomputadores usan encaminamiento segmentado (wormhole routing), de ahí que hayamos hecho las modificaciones oportunas al algoritmo para que contemple esta nueva situación.

La idea básica es la siguiente: cuando los mensajes que llegan a un determinado destino han soportado un importante retraso, el algoritmo intentará poner este destino lo más cerca posible del lugar que origina tales retrasos, intercambiando su posición con su vecino más próximo a la zona de conflicto.

El algoritmo trabaja, pues, de la siguiente forma:

1. Cada vez que un mensaje llega a su destino, se registra la información sobre la contención. Se tendrá en cuenta un límite para este valor (*Umbral_2*), para prevenir que un solo mensaje que ha estado mucho tiempo parado pueda desencadenar el proceso de reconfiguración. Más aún, se requiere un número mínimo de mensajes (*Umbral_3*) para que se active el proceso de reconfiguración.

2. El algoritmo chequea el estado del nodo cada vez que el número de mensajes recibidos alcanza la cota *Umbral_1*.

3. Si la contención por el canal es mayor que la suma en el resto y mayor que *Umbral_4*, el nodo envía un mensaje al nodo unido a él por ese canal para indicarle la conveniencia de intercambiar sus posiciones.

4. Si el cambio es adecuado para ambos, comienza el protocolo de reconfiguración que se describe más adelante.

De forma esquemática, y para un nodo dado i, el algoritmo que se ejecuta cada vez que se recibe un mensaje es:

```
registrar_informacion(mensaje);
actualizar(contador_mensajes);
if (contador_mensajes % umbral_1 == 0)
   {
     procesar_informacion(continuar);
     if (continuar) chequear_cambio(cambiar);
     if (cambiar) reconfigurar(nodo_i,vecino);
   }
```

donde:

contador_mensajes, mide el número total de mensajes recibidos por ese nodo,

umbral_1, es el intervalo de tiempo entre cada evaluación de la contención en la red,

procesar_informacion, evalúa la contención encontrada por los mensajes que llegan al nodo por un canal dado. Se tienen en cuenta para ello dos umbrales: la contención mínima (*umbral_2*) y el número mínimo de mensajes que han de llegar para desencadenar el proceso de reconfiguración (*umbral_3*),

chequear_cambio, verifica la idoneidad del cambio. En esta verificación interviene de un lado la situación del nodo vecino y de otro un límite para el valor mínimo de la contención (*umbral_4*),

cambiar, es una variable booleana que recoge el resultado del proceso anterior,

reconfigurar, es un procedimiento que inicia el protocolo de reconfiguración.

El algoritmo, como se puede deducir de lo anterior, tiene las siguientes características:

- Preserva la topología: después de un cambio la red mantiene la misma topología.
- Está basado en la contención en la red: un nodo determina la conveniencia de cambiar en base a la información sobre la pérdida de ancho de banda en la red.
- Produce alteración pequeña: un nodo puede intercambiar su posición sólo con alguno de sus vecinos.
- Usa cuatro umbrales a lo largo del proceso de reconfiguración.

Finalmente, nuestro algoritmo provoca un número reducido de cambios en la red. Esta característica es importante debido al tiempo que se invierte en la realización de cada uno de los cambios.

A continuación, proponemos la función de coste en la cual se basa el algoritmo de reconfiguración.

2.2 Función de coste

Un nodo determina la conveniencia de cambiar de posición teniendo en cuenta información sobre la contención en la red. Dicha información le llega a través de los mensajes que recibe. Cada nodo registra el tiempo durante el cual un mensaje ha estado parado en la red y los canales que ha estado ocupando durante ese tiempo. Con ambos factores podemos estimar la pérdida de ancho de banda en la red debido al problema de la congestión.

Así, para un nodo D dado y uno cualquiera de sus canales j, la función de coste toma la siguiente expresión:

$$FC_j = \sum_{mensajes} \left(\sum_{N=S}^{D} t_b * c_o \right)$$

donde t_b es el tiempo que un mensaje con destino D ha estado bloqueado a lo largo de su camino, desde su salida del nodo origen S, y c_o es el número de canales bloqueados durante t_b.

2.3 El protocolo para la reconfiguración

Se ha elegido un control distribuido para la reconfiguración dinámica de la red de interconexión. Cada nodo es responsable de controlar y desencadenar el proceso de reconfiguración en la parte lógica. Los aspectos físicos de la misma son centralizados en un solo nodo o controlador.

Los principales pasos del protocolo se detallan a continuación:

1. Cuando un par de nodos decide que es conveniente intercambiar sus posiciones, informan a todos sus nodos vecinos de sus intenciones, indicándoles que no les envíen más mensajes.

2. El nodo que propone el cambio envía los datos de la reconfiguración al nodo que controla el switch.

3. Este nodo controlador modifica la topología de la red, adaptándola a las nuevas circunstancias.

4. El controlador del switch envía a todos los nodos la nueva situación de los dos nodos implicados en el cambio. Estos reestablecen la comunicación con sus vecinos.

Este protocolo es fácilmente implementable haciendo uso, por ejemplo, de un bus, y sin añadir demasiado tráfico en la red (ver más adelante la figura 5).

3 Sistema de transputers reconfigurable

En los últimos años está aumentando el interés por los sistemas dinámicamente reconfigurables y hay varios de ellos en estado experimental [9, 11]. Es relativamente fácil encontrar en el mercado componentes apropiados para construir este tipo de sistemas, aunque no lo es tanto obtener sistemas reconfigurables que sean eficientes, ya que se necesita una buena solución arquitectónica así como una gran rapidez en la tecnología disponible.

La llegada de los encaminadores C104 para la nueva generación de transputers T9000 ha estimulado el estudio de este tipo de sistemas. El chip de encaminamiento ST C104 contiene un crossbar 32 x 32 no bloqueante, es decir, las 32 entradas se pueden encaminar al mismo tiempo hacia las 32 salidas [8]. Este encaminador usa la técnica de control de flujo wormhole, que, como ya se explicó anteriormente, minimiza los retrasos de encaminamiento debido a que la cabecera de un paquete se encamina en cuanto llega, pudiendo empezar a salir del encaminador aunque el resto del paquete aún esté entrando.

La técnica wormhole requiere una estrategia de encaminamiento eficiente para decidir por qué canal debe sacarse un paquete. El encaminador C104 usa un esquema de encaminamiento basado en etiquetas (interval labelling), donde a cada vía de salida del C104 se le asigna un rango o intervalo. Este intervalo contiene los números asignados a todos los nodos terminales (por ejemplo microprocesadores, puertas a otras redes, periféricos, etc) que son accesibles a través de ese canal. Cuando llega un paquete al encaminador C104 se compara la etiqueta de la cabecera del paquete donde se indica cual es su destino, con el conjunto de intervalos, y aquel intervalo en que está incluido dicho destino nos da el canal de salida que debe seguir. Podemos ver un ejemplo de esto en la figura 2. En él, la etiqueta de destino del paquete indica que su destino es el nodo 123 y consultando la tabla se obtiene que el canal de salida debe ser el número 3.

Aunque el encaminador C104 no restringe el tamaño de los paquetes, el transputer T9000 si que lo hace, y así un paquete puede contener hasta 32 bytes de datos. Si un

123

cabecera

Intervalos	Canales
$0 \leq n < 16$	0
$16 \leq n < 32$	1
$32 \leq n < 64$	2
$64 \leq n < 128$	3
$128 \leq n < 256$	4

$\rightarrow$ Envía el paquete por el canal 3

Figura 2: Un ejemplo de encaminamiento por intervalos

mensaje es mayor de 32 bytes se subdivide en varios paquetes. El encaminador C104 permite el encaminamiento determinista y el adaptativo. En el primero, la ruta que sigue un mensaje sólo depende de los nodos origen y destino, mientras que en el encaminamiento adaptativo la ruta que sigue un mensaje puede variar de una vez a otra dependiendo del estado de la red.

El encapsulado IMS T9000 [8] es un transputer de la segunda generación. Tiene un procesador superescalar, un planificador por hardware, 16 K bytes de memoria cache y un procesador de comunicaciones autónomo.

El planificador del transputer T9000 se encarga de la creación y ejecución de cualquier número de procesos de forma concurrente. Dichos procesos se comunican por intercambio de mensajes a través de canales punto a punto. Estos canales son unidireccionales, y el paso de mensajes es sincronizado y sin almacenamiento intermedio.

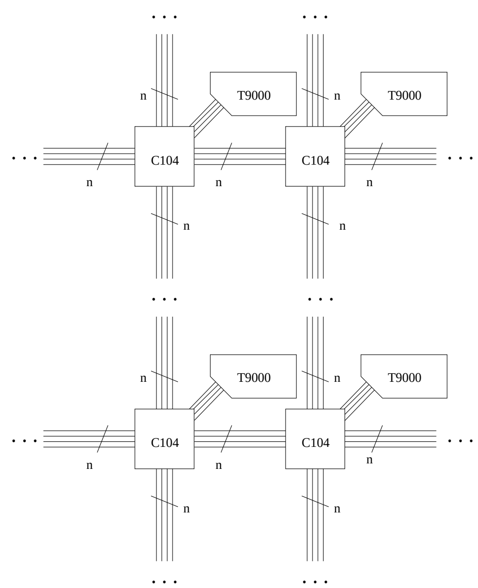

Figura 3: Una red basada en el transputer T9000 y el encaminador C104

El sistema de comunicaciones del transputer T9000 tiene cuatro canales, cada uno de los cuales puede conectarse directamente a otro canal de otro transputer, o a través de una red de encaminamiento a otros transputers.

La pareja formada por el transputer T9000 y el encaminador C104 nos permite la construcción de grandes sistemas distribuidos (figura 3).

3.1 Arquitectura general del sistema

El sistema de transputers dinámicamente reconfigurables (figura 4) consiste de varios nodos formados por el conjunto del transputer T9000 y el encaminador C104 (figura 3), un conmutador de enlaces que permiten la comunicación entre los nodos, una línea de control y un transputer de control que supervisa la configuración de los enlaces.

El conmutador de enlaces puede ser una red de barras cruzadas multi-entrada o bien una red multinivel formada por muchos conmutadores elementales de dos entradas [10].

El número de nodos conectados al conmutador depende del número de conexiones simultáneas que el conmutador es capaz de establecer. Así, para una topología toro 2D, donde cada nodo tiene cuatro conexiones, una con cada uno de sus vecinos, podemos conectar $n/4$ nodos, donde n es el número de conexiones simultáneas que el conmutador es capaz de establecer.

Por ejemplo para un conmutador $256x256$ podríamos tener hasta 64 nodos formando una topología toro 2D. Para implementar un toro 2D, el valor de k en la figura 4 debe ser de cuatro enlaces. Para una topología hipercubo con N nodos, donde $N = k^n$, siendo n el número de dimensiones y k el número de nodos por dimensión, cada nodo tiene n conexiones, una con cada uno de sus vecinos en cada dimensión. Así, para un conmutador mxm podemos tener hasta m/n nodos.

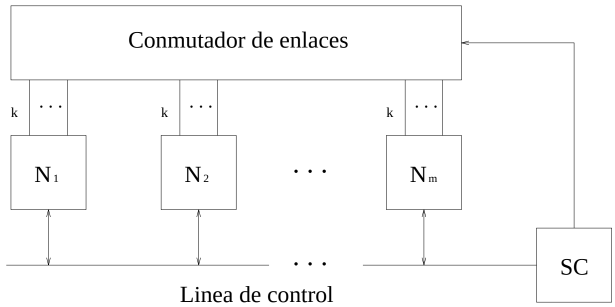

Figura 4: Estructura del sistema para la reconfiguración dinámica

Aunque actualmente el tamaño de los conmutadores está limitado, podemos formar redes con un mayor número de nodos replicando la estructura de la figura 5 e introduciendo un conmutador global [10]. En dicha figura puede verse como el control del bus se hace por fases usando otro controlador para manejar el conjunto de buses.

Los nodos no se conectan directamente al bus de control, sino que usamos un manejador especial que nos permite agilizar el bus y hacer su funcionamiento transparente a los nodos, que se limitan a pedir las comunicaciones, despreocupándose de como éstas se llevan a cabo.

Nosotros proponemos como manejador para el bus de control el desarrollado en el instituto IRESTE de la Universidad de Nantes llamado TRANSBUS [3]. El TRANSBUS permite una comunicación eficiente entre los nodos y el transputer de control del bus.

El TRANSBUS (figura 6) actúa como un interface entre los nodos y el bus. Cada nodo se conecta al TRANSBUS por una única línea, mientras que el TRANSBUS tiene cuatro líneas: una para los datos, otra para los reconocimientos y dos para la entrada y salida del token, que permite a los nodos poner información en el bus sin colisiones.

Si un nodo va a comunicarse por el bus, manda la dirección de destino del mensaje a su TRANSBUS, donde se almacena la dirección a la espera de recibir el token, y cuando éste llega, la dirección del mensaje se envía por la línea de datos del bus y se espera a que los otros controladores manden sus reconocimientos a través de la línea de reconocimiento.

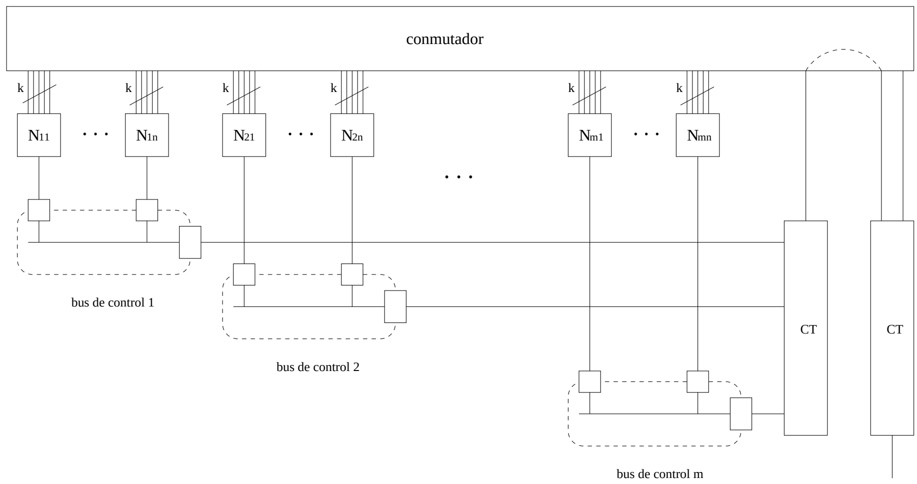

Figura 5: Estructura general de una arquitectura reconfigurable basada en bus serie

La dirección de destino se lee y decodifica por todos los controladores de los TRANSBUS situados en el bus, y sólo al nodo cuya dirección coincide con la indicada (o a todos en caso de un envío general que se hace poniendo una dirección de destino nula) se le transmite el resto del mensaje desde el controlador.

Cuando un nodo recibe un byte, éste hace un reconocimiento por el enlace del nodo al controlador del TRANSBUS, y luego éste retransmite dicho reconocimiento al nodo origen a través de la línea de reconocimiento del bus de control.

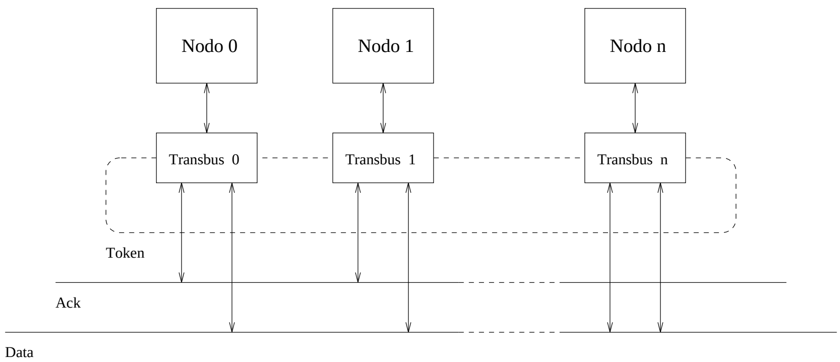

Figura 6: Interconexión de los nodos mediante el controlador TRANSBUS

4 Conclusiones

En este artículo mostramos un nuevo algoritmo para la reconfiguración dinámica de sistemas multi-transputer, así como las ventajas de usar este tipo de sistemas en el desarrollo de una gran variedad de aplicaciones.

Incluimos la arquitectura de un sistema de transputers reconfigurable basado en bus serie y que usa el transputer T9000 y el encaminador C104. El control de la reconfiguración está basado en el intercambio de mensajes a través del bus, el cual conecta el nodo controlador del switch con el resto de nodos.

La evaluación se está realizando con un entorno de programación y simulación desarrollado en el seno de nuestro grupo, el cual nos permite chequear el comportamiento del algoritmo para la reconfiguración sobre muy diversas configuraciones y para diferentes tipos de carga. Para ello se varían los parámetros habitualmente utilizados en la evaluación de sistemas masivamente paralelos y que tienen que ver con la topología de la red de interconexión, la técnica de control de flujo y el algoritmo de encaminamiento de los mensajes.

En cuanto a las cargas que estamos utilizando hay que indicar que nuestro entorno permite manejar dos tipos básicos: cargas que son el resultado de la ejecución simulada de algoritmos paralelos, y por otro lado cargas obtenidas a partir de distribuciones típicas de mensajes y sus destinos. Para el primer tipo hacemos uso de algoritmos numéricos muy utilizados, como triangularización de matrices dispersas. Las cargas predeterminadas más usuales corresponden a distribuciones uniformes, exponenciales o esféricas, a las que unimos casos especiales como las generadas por patrones de comunicaciones con puntos calientes.

Por último, indicar que se sigue trabajando en el tema, y se está haciendo en varios frentes, conducentes todos ellos a chequear el grado de bondad introducido por la técnica sometida a estudio. Así, se continúa perfeccionando el algoritmo utilizado, ajustando la función de coste y los diferentes umbrales que intervienen. Por otra parte, se trabaja en el desarrollo de diversas alternativas a la implementación física de este tipo de redes: basadas en crossbar, con bus paralelo, redes multietapa, etc. Una vez completado dicho estudio, estaremos en condiciones de aportar unos resultados totalmente precisos que determinen las verdaderas posibilidades de este tipo de redes.

Referencias

[1] A. Agarwal. *Limits on interconnection network performance.* IEEE Transactions on Parallel and Distributed Systems, Vol. 2, No. 4, pp. 398–412, October 1991.

[2] A.A. Chien. *A cost and speed model for k-ary n-cube wormhole routers.* In Proc. of Hot Interconnects'93, August 1993.

[3] J.P. Calvez, O. Pasquier. *A transputer interconnection bus for hard real-time systems.* In Proc. TRANSPUTER'92 Conference, Besancon, France, May 1992.

[4] W.J. Dally, C.L. Seitz. *Deadlock-free message-routing in multiprocessor interconnection networks.* IEEE Transactions on Computers, Vol. C-36, No. 5, pp. 547–553, May 1987.

[5] Ch. Fraboul, J.Y. Rousselot, P. Siron. *Software tools for developing programs on a reconfigurable parallel architecture.* In D. Grassilloud, J.C. Grossetie, editors, *Computing with Parallel Architectures: T. Node*, pp. 101–110. Kluwer Academic Publishers, 1991.

[6] J.M. García, J. Duato. *Dynamic reconfiguration of multicomputer networks: Limitations and tradeoffs.* In P. Milligan, A.Nuñez, editors, *Euromicro Workshop on Parallel and Distributed Processing*, pp. 317–323. IEEE Computer Society Press, 1993.

[7] P. Kermani, L. Kleinrock. *Virtual cut-through: a new computer communication switching technique.* Computer Networks, Vol. 3, pp. 267–286, 1979.

[8] M.D. May, P.W. Thompson, P.H. Welch, et al. *Networks, Routers & Transputers: Function, performance and application.* IOS Press, Amsterdam, 1993.

[9] A. Murta. *Active reconfiguration of transputer links for network-wide occam communication.* Department of Computer Science, University of Manchester, 1992.

[10] M. Tudruj, T. Kalinowski. *Multi-transputer systems with dynamic link connection switching controlled through a serial bus.* In World Transputer Congress '93, pp. 803–818, Aachen, September 1993.

[11] M. Tudruj, M. Thor. *Implementation of an experimental multi-transputer system with dynamically reconfigurable topology.* In 8-th Symposium on Microcomputer and Microtransputer Applications, pp. 345–356, Budapest, October 1994.