

Reconfiguración dinámica en NOWs: un paso hacia la garantía de QoS

F.J. Alfaro, A. Bermúdez, R. Casado, P.J. García, J. Duato, F.J. Quiles, J.L. Sánchez

Abstract— Las actuales técnicas de reconfiguración empleadas en NOWs no garantizan la QoS requerida por ciertas aplicaciones, como aquellas que soportan tráfico multimedia. El presente documento describe nuestros estudios dirigidos a desarrollar algoritmos de reconfiguración que no degraden en exceso las prestaciones de la red. Nuestros algoritmos se basan en distintas técnicas de control de bloqueos, como evitación y recuperación, y se adaptan a redes que empleen encaminamiento fuente o distribuido.

Keywords— Redes de estaciones de trabajo, reconfiguración dinámica, gestión de bloqueos, calidad de servicio

I. INTRODUCCIÓN

Hace dos años, el grupo de *Redes y Arquitecturas de Altas Prestaciones (RAAP)* inició una nueva línea de investigación destinada a desarrollar algoritmos de reconfiguración de NOWs. Nuestro objetivo era (y es) solventar los inconvenientes que presentan las técnicas de reconfiguración, especialmente cuando el tráfico soportado por la red requiere cierta calidad de servicio (QoS).

En las VIII Jornadas de Paralelismo [1] se establecieron las premisas que han marcado esta línea de investigación. En las IX Jornadas de Paralelismo [2] se presentaron los primeros resultados obtenidos: una metodología de reconfiguración dinámica de la red de interconexión. Esta metodología, a diferencia de las técnicas anteriores, no requiere detener el tráfico generado por las aplicaciones durante su ejecución. Las preguntas que surgieron en aquel momento fueron fundamentalmente de dos tipos, ¿funciona realmente el algoritmo desarrollado?, ¿proporciona mejores resultados que los obtenidos por las técnicas anteriores?.

Este documento tiene dos propósitos. El primero

A. Bermúdez, R. Casado, P. J. García, F.J. Quiles, J.L. Sánchez pertenecen al Departamento de Informática de la Universidad de Castilla-La Mancha, (Escuela Politécnica Superior, Campus Universitario s/n, 02071 Albacete). e-mail: {abermu,rcasado,pgarcia,paco,jsanchez}@info-ab.uclm.es

F.J. Alfaro pertenece al Departamento de Ingeniería y Tecnología de Computadores de la Universidad de Murcia, (Facultad de Informática, Campus de Espinardo, 30071 Murcia). e-mail: falfaro@dittec.um.es

J. Duato pertenece al Departamento de Informática de Sistemas y Computadores de la Universidad Politécnica de Valencia (Facultad de Informática, Camino de Vera s/n, 46071 Valencia). e-mail: jduato@gap.upv.es

Este trabajo está parcialmente soportado por el proyecto CICYT TIC97-0897-C04

de ellos es presentar brevemente las nuevas líneas de investigación surgidas a partir de la original y sobre las que diversos investigadores están desarrollando su tesis doctoral. El segundo es contestar aquellas preguntas que quedaron sin respuesta, presentando los primeros resultados de simulación obtenidos.

El resto del documento está organizado de la siguiente forma: la sección II presenta, de una forma general, la necesidad de la reconfiguración en NOWs; las secciones III y IV plantean la problemática asociada a la reconfiguración de la red de interconexión en redes que empleen encaminamiento distribuido y encaminamiento fuente respectivamente; la sección V describe las características que debería reunir un protocolo de nivel superior capaz de garantizar la QoS requerida.

II. RECONFIGURACIÓN EN NOWs

Las NOWs actuales, tales como Autonet [3], Myrinet [4] y Servernet [5] incorporan técnicas propias de las redes de interconexión de computadores paralelos. Algunos ejemplos son la utilización de enlaces punto a punto entre conmutadores o técnicas de conmutación segmentadas. Este tipo de redes también ha heredado características de las redes locales convencionales, como su flexibilidad en el cableado y la variabilidad en la topología de la red.

Estas propiedades provocan la aparición de ciertos problemas relacionados con la configuración de la topología y el encaminamiento de paquetes. En concreto, la topología puede cambiar debido a la activación y desactivación de conmutadores y terminales, el reajuste de sus conexiones o fallos en los componentes. En estos casos, con el fin de proporcionar una alta disponibilidad del sistema, un *algoritmo de reconfiguración* debe actualizar las tablas de encaminamiento para que la comunicación siga siendo posible entre los diferentes componentes de la red.

Cada tipo de red puede adoptar una filosofía diferente a la hora de implementar su mecanismo de reconfiguración. A pesar de estas diferencias, en todos los casos se observa una degradación de las prestaciones de la red de interconexión. Aunque esta degradación es inevitable, es posible reducirla optimizando el proceso de reconfiguración.

III. RECONFIGURACIÓN EN REDES CON ENCAMINAMIENTO DISTRIBUIDO

A. Implementaciones actuales: Autonet

Un ejemplo representativo de red con encaminamiento distribuido es la red Autonet. Autonet [3] dispone de conmutadores “inteligentes” que incorporan un procesador interno (*MC68000*) que monitoriza completamente su funcionamiento. Estos conmutadores implementan un complejo y robusto mecanismo de detección de cambios. Cuando se produce un cambio significativo en la topología, Autonet ejecuta un proceso de reconfiguración distribuido que se extiende a toda la red y actualiza la tabla de encaminamiento en cada conmutador.

A grandes rasgos, podemos identificar tres fases en el proceso de reconfiguración. Una primera fase (*distribuida*) tiene por objeto generar un árbol de expansión de la red (*spanning tree*), que se utiliza como punto de partida para obtener un grafo dirigido acíclico que será usado en la última fase.

En una segunda fase (*centralizada*), la raíz del árbol de expansión sintetiza toda la información sobre la nueva topología, recogida durante la primera fase del proceso, y la difunde al resto de conmutadores, valiéndose del árbol obtenido.

La tercera y última fase consiste en la construcción de las tablas de encaminamiento, labor que realiza cada conmutador una vez que dispone del grafo dirigido. Las tablas se crean aplicando a este grafo el algoritmo *up*/down** [6].

Cuando la red de interconexión sufre una modificación en su topología, ésta suele afectar a una zona reducida de la misma comparada con el tamaño total de la red. Un problema asociado al algoritmo descrito es que realiza una reestructuración completa de la topología, independientemente del tamaño de la zona realmente afectada. La consecuencia es que el tiempo que tarda en ejecutarse la reconfiguración es siempre el máximo posible. Los métodos que se comportan de esta forma los denominamos *técnicas de reconfiguración total*.

La inconsistencia que sufren temporalmente las tablas de encaminamiento mientras se realiza el proceso de reconfiguración puede provocar la aparición de situaciones de bloqueo entre paquetes de usuario. Con el fin de evitarlos, el tráfico generado por las aplicaciones es detenido. Cuando el proceso termina, se reactiva su circulación. En estas situaciones decimos que se ha producido una reconfiguración *estática* de la red de interconexión. El mecanismo incorporado en Autonet responde a esta definición.

La detención de tráfico producida por una reconfiguración estática conlleva una degradación de las prestaciones de la red. La figura 1 (parte superior) muestra el efecto que produce una reconfiguración

estática en la latencia instantánea del sistema. Puede apreciarse como la latencia se incrementa en varios órdenes de magnitud durante el proceso de reconfiguración. Un gran número de aplicaciones distribuidas que proporcionan una garantía de servicio (QoS) a sus usuarios podrían verse seriamente afectadas por este tipo de reconfiguración. La transmisión de vídeo bajo demanda podría ser un buen ejemplo, pues el flujo de información generado podría verse afectado por el incremento de latencia hasta el punto de no poder satisfacer las condiciones temporales impuestas por la transmisión. Muchos organismos (como el ministerio de defensa de los EEUU) rehusan sustituir sus multicomputadores por este tipo de redes en tanto este problema no haya sido solucionado.

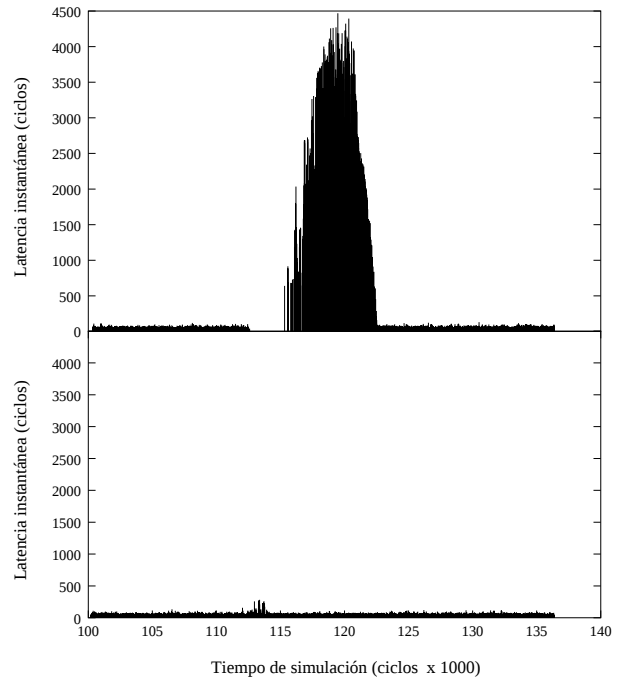


Fig. 1. Resultados de simulación para una red con encaminamiento distribuido. Parte superior: reconfiguración estática. Parte inferior: reconfiguración dinámica.

Este último inconveniente aconseja abordar la reconfiguración de la red desde una nueva perspectiva: *realizar la reconfiguración sin que ello implique una paralización del tráfico de usuario*, aplicando lo que hemos denominado *técnicas de reconfiguración dinámica*. El objetivo es reducir los efectos negativos que la reconfiguración produce directamente sobre la latencia de la red, e indirectamente sobre el grado de satisfacción del usuario. El resto de esta sección describe dos formas de enfocar la reconfiguración dinámica basadas en distintas técnicas de control de bloqueos.

B. Reconfiguración dinámica con técnicas de evitación de bloqueos

El algoritmo de reconfiguración dinámica basado en encaminamiento distribuido y con evitación de bloqueos se describe detalladamente en [7], [8], por lo que en este trabajo sólo se comentan brevemente sus propiedades.

La alternativa que proponemos a la *reconfiguración total* de Autonet es la *reconfiguración parcial* de la red: nuestro algoritmo aprovecha la zona útil de la configuración existente. El resultado es que la mayor parte de los cambios de topología que se producen se asimilan de forma ágil y rápida. La figura 2 muestra una red en la que se produce un cambio de topología. Las figuras 3 y 4 muestran las zonas de la red afectadas por una reconfiguración parcial y total, respectivamente.

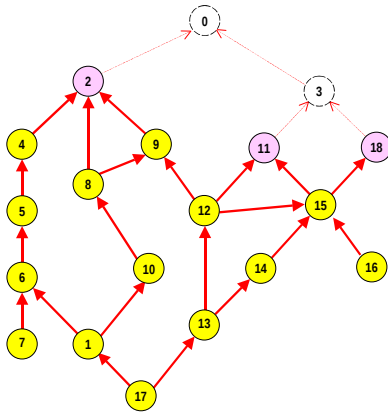


Fig. 2. Ejemplo de cambio en la topología. La desactivación de los nodos 0 y 3 dispara un proceso de reconfiguración.

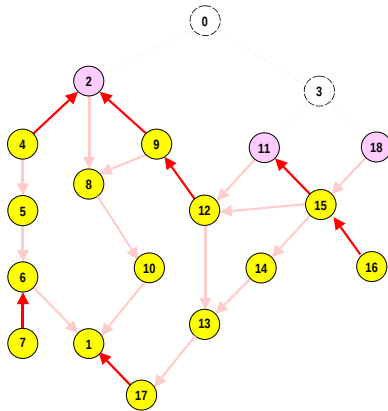


Fig. 3. Grafo tras una reconfiguración total. Los enlaces invertidos se muestran en color claro.

Por otra parte, nuestro algoritmo realiza la reconfiguración de la red de interconexión de forma dinámica, es decir, el tráfico de aplicación se mantiene durante el proceso. El resultado es que

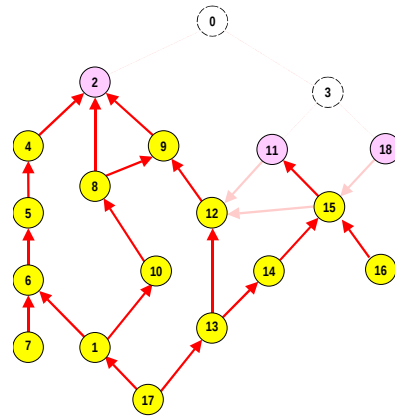


Fig. 4. Grafo tras una reconfiguración parcial. La cantidad de enlaces invertidos es menor que en la figura 3

la degradación de prestaciones asociada a la reconfiguración resulta atenuada, tal y como se muestra en la figura 1 (parte inferior).

Si no se detiene el tráfico, surge la posibilidad de que aparezcan bloqueos debidos a la interacción de múltiples funciones de encaminamiento correspondientes a diversas configuraciones. Por tanto, es necesario que el mecanismo de control de bloqueos actúe también durante el proceso de reconfiguración.

Nuestro algoritmo de reconfiguración incorpora una técnica de evitación de bloqueos basada en prevenir la formación de ciclos en el grafo de dependencia de canales [9]. Para conseguirlo se recurre a una serie de modificaciones parciales y progresivas de las tablas de encaminamiento.

Dentro de esta línea, estamos desarrollando el mismo protocolo de reconfiguración sobre encaminamiento adaptativo, evitando ciclos en el grafo de dependencias de canales extendido [10].

C. Reconfiguración dinámica con técnicas de recuperación de bloqueos

El empleo de una técnica de evitación de bloqueos aumenta la complejidad del protocolo de reconfiguración. Esto es debido a que es imprescindible una modificación parcial y progresiva de las tablas de encaminamiento.

Como alternativa para reducir la complejidad del protocolo de reconfiguración podría pensarse en la combinación de un *mecanismo de detección de bloqueos* eficaz con el posterior uso de una *técnica de recuperación* de los mismos.

Un mecanismo de detección simple, pero ineficiente, es el uso de “timeouts”. Así, se puede deducir que un determinado paquete está bloqueado si ha transcurrido demasiado tiempo desde que fuera enviado, o bien si lleva demasiado tiempo esperando para ser encaminado a la entrada de un conmutador.

Un método mucho más eficaz -y no mucho más

complejo- es el *FC3D (Flow Control-based Distributed Deadlock Detection)* [11]. Este método se basa en detectar el bloqueo a partir de la interpretación de los mensajes de control de flujo que intercambian los conmutadores.

En cuanto a las técnicas de recuperación, nos encontramos con dos alternativas: usar una *técnica regresiva*, o bien emplear una *técnica progresiva*. La recuperación regresiva se basa en el descarte de uno o más de los paquetes involucrados en el ciclo, con el objetivo de romperlo. Estos mecanismos actúan junto con un protocolo de nivel superior que detecte dicho descarte mediante un mecanismo de reconocimiento/reenvío.

Por su parte, las técnicas de recuperación progresiva, al contrario que las anteriores, no descartan ningún paquete, pero añaden cierta complejidad. Esta complejidad puede reflejarse en el hardware, como es el caso del encaminador *Disha* [12], o bien en el software [13]. La idea en ambos casos es también extraer del ciclo alguno de los paquetes bloqueados. Para ello, *Disha* añade al conmutador una cola especial hacia donde es dirigido dicho paquete. En la técnica de recuperación basada en software el paquete es absorbido por el procesador, para ser reinyectado una vez que la situación de bloqueo ha sido eliminada.

Independientemente de la técnica de detección y recuperación aplicada, parece lógico pensar que el uso de técnicas de recuperación de bloqueos simplificaría sensiblemente el protocolo de reconfiguración. De este modo se ofrecería una solución a los bloqueos sin detener el tráfico de aplicación, a costa de la mencionada sobrecarga hardware/software. Una de nuestras líneas es determinar las ventajas de este nuevo enfoque.

IV. RECONFIGURACIÓN EN REDES CON ENCAMINAMIENTO FUENTE

A. Encaminamiento fuente

En la sección anterior hemos abordado la problemática de la reconfiguración para redes como Autonet, que emplean encaminamiento distribuido. Ahora nos centraremos en redes que emplean *encaminamiento fuente*. En estas redes las decisiones para el encaminamiento de un paquete se toman únicamente en el terminal emisor. Las rutas hacia otros terminales se determinan completamente en el origen y cada paquete contiene información suficiente para llegar a su destino siguiendo la ruta definida. El papel de los conmutadores en este tipo de redes es pasivo por cuanto no pueden producir ni “reenaminar” paquetes, limitándose a darles salida por el puerto indicado en el propio paquete. Dicho de otro modo: los conmutadores carecen de tablas de encaminamiento, que sólo están presentes en los ter-

minales.

En redes que empleen este tipo de encaminamiento, las limitaciones de los conmutadores pasivos obligan a plantear el problema de la reconfiguración de la red bajo premisas distintas a las expuestas hasta ahora. Por ejemplo, un conmutador pasivo es incapaz de detectar un cambio en la topología y disparar un proceso de reconfiguración. Esta responsabilidad recae en los terminales.

Entre las actuales NOWs que implementan esta técnica de encaminamiento se encuentra Myrinet, desarrollada por Myricom [4].

B. Implementaciones actuales: Myrinet

Myrinet implementa un mecanismo de reconfiguración centralizado. Un único terminal de la red (denominado *mapper*) se encarga de explorar la topología para detectar cualquier cambio, confeccionar un nuevo mapa de la red y comunicarlo al resto de los terminales para que actualicen sus tablas de encaminamiento (según el algoritmo *up*/down**). Los conmutadores Myrinet, ajustándose al modelo de conmutadores pasivos, carecen de dichas tablas, e incluso de dirección (*UID, unique identifier*) propia. La arquitectura de Myrinet complica la tarea del *mapper* notablemente. Por ejemplo, la carencia de *UID* propio para los conmutadores obliga a complejas estrategias para evitar que un mismo conmutador sea duplicado en el mapa [14].

En Myrinet, no se detiene el tráfico de usuario durante el proceso de reconfiguración. La solución prevista para los posibles bloqueos originados durante la actualización de las tablas se basa en eliminar los paquetes cuyo tiempo de transmisión supere un determinado timeout. De esta forma ningún paquete puede estar bloqueado indefinidamente. En otras palabras, Myrinet emplea una estrategia de recuperación de bloqueos durante el proceso de reconfiguración, en lugar de evitarlos. Podríamos hablar, por tanto, de una técnica de *reconfiguración dinámica con recuperación de bloqueos*.

Como se menciona en la sección anterior, este mecanismo implica la necesidad de retransmitir la información que se pierde al romper bloqueos. Myrinet no define el modo en que esta retransmisión debe efectuarse (al menos hasta la reciente aparición de Myricom GM [4]), dejando esta responsabilidad a protocolos superiores, como TCP. La presencia de este protocolo en Myrinet produce una sobrecarga software que degrada las prestaciones de la red [15] hasta extremos que pueden considerarse inaceptables para una red de altas prestaciones. Tanto es así que Myricom ha abandonado este planteamiento inicial, y las últimas implementaciones del mecanismo de reconfiguración de Myrinet (Myricom GM *mapper*) no permiten la circulación de paquetes de

usuario durante el proceso de reconfiguración. En definitiva, se evoluciona hacia una técnica de *reconfiguración estática* similar a la de Autonet.

C. Reconfiguración dinámica con técnicas de evitación de bloqueos en Myrinet

A la vista de los resultados obtenidos por el algoritmo de reconfiguración dinámica con evitación de bloqueos presentado en la sección III nos planteamos la posibilidad de adaptarlo a redes con encaminamiento fuente, y en concreto a Myrinet. Esta red presenta la ventaja de que todas las funciones relativas al encaminamiento y reconfiguración se implementan mediante un programa de control (MCP, Myrinet Control Program [4]) que se ejecuta en el interfaz de red de cada terminal. Myricom permite a sus clientes el acceso al código de dicho programa para adaptarlo a sus necesidades. Por tanto, basta con modificar convenientemente el MCP para cambiar la técnica de reconfiguración de la red. Es por esto que Myrinet se presenta como una plataforma de prueba ideal donde implementar un algoritmo de reconfiguración acorde con nuestros planteamientos.

A pesar de las diferencias existentes entre redes que emplean encaminamiento distribuido y encaminamiento fuente, las líneas principales del algoritmo de reconfiguración dinámica con evitación de bloqueos desarrollado para las primeras siguen siendo válidas para las segundas. Así, puede intuirse que la evitación de bloqueos durante el proceso de reconfiguración también debe basarse en que la actualización de las tablas de encaminamiento se realice de forma parcial y progresiva. En este caso debe tenerse muy en cuenta que los paquetes no pueden ser reencaminados en los conmutadores, y que el mapper, como responsable del proceso de reconfiguración, debe ser consciente en cada momento del estado del tráfico en determinados enlaces. También emplearemos grafos de dependencia de canales como herramienta para verificar la ausencia de bloqueos durante el proceso.

En general, la adaptación de cualquier aspecto del algoritmo de reconfiguración dinámica a redes con encaminamiento fuente debe ser posible si se tienen en cuenta las características de estas últimas. Nuestra intención es implementar este algoritmo adaptado en Myrinet, modificando su programa de control. Esto nos permitiría calibrar el impacto del proceso de reconfiguración en las prestaciones de la red. Un análisis comparativo de los resultados permitiría a su vez comprobar la bondad del algoritmo de reconfiguración dinámica con evitación de bloqueos frente a otras técnicas de reconfiguración.

V. GARANTÍA DE QoS EN NOWs

Algunas aplicaciones distribuidas de tiempo real, requieren condiciones muy estrictas en su tráfico [16], [17] con rigurosas limitaciones en parámetros como la varianza del retraso de las células ATM (CDV, Cell Delay Variance), el máximo retraso entre células (CTD, Cell Transfer Delay), el porcentaje de células perdidas (CLR, Cell Lost Ratio), etc.

Las aplicaciones multimedia distribuidas (videoconferencia, reuniones virtuales, vídeo bajo demanda, etc), tienen también requerimientos, aunque menos estrictos, en cuanto a QoS. En estos casos es necesario que la red pueda proporcionar esos requerimientos mínimos de QoS sin los cuales dichas aplicaciones no llegarán a utilizarse. Este tipo de aplicaciones suelen estar orientadas a la conexión, es decir, primero se establece la conexión entre las entidades, luego se produce el trasiego de información, y finalmente se libera la conexión. La conexión se establece si es posible garantizar unos mínimos requerimientos durante la misma.

Las necesidades de QoS en aplicaciones multimedia distribuidas suelen expresarse en términos de:

- un ancho de banda mínimo garantizado que permita que al nodo destino pueda llegar un cierto número mínimo de paquetes por unidad de tiempo.
- un retraso entre llegadas (*jitter*), garantizado. En este tipo de aplicaciones puede ser fundamental que el siguiente frame de vídeo/audio llegue al destino en un tiempo determinado.

En las redes de área local, el tráfico generado por aplicaciones con necesidades de QoS debe coexistir con *tráfico al mejor intento*, que no requiere garantías de ningún tipo. En estas redes la QoS suele soportarse en capas superiores (la capa de transporte), que introducen una considerable sobrecarga software. En redes de altas prestaciones el retraso introducido por dichas capas podría resultar inaceptable, y deben considerarse nuevos métodos para proporcionar esta QoS.

Trabajos recientes han abordado esta cuestión y tres son las alternativas que se presentan:

- Crear dos subredes separadas [18] añadiendo enlaces a la topología: una para soportar exclusivamente el tráfico con necesidades de QoS y la otra dedicada al tráfico al mejor intento.
- Crear un marco “síncrono” dentro de la red “asíncrona” [18], [19]. Una multiplexación en el tiempo permite que varias conexiones usen, en instantes distintos, los mismos canales sin interferencias. Para ello es imprescindible una correcta asignación de slots de tiempo a cada conexión.
- Usar gran cantidad de canales virtuales que permitan atender simultáneamente muchas conexiones y tráfico al mejor intento [20], [18].

La reconfiguración estática descrita anteriormente

(sección III) no permite ningún tipo de QoS al detener el tráfico de usuario durante un proceso de reconfiguración. También es usual que cuando ocurre una reconfiguración en la red los nodos tengan que descartar muchos paquetes. Esto puede solucionarse con la presencia de una capa software superior que reenvíe los paquetes descartados, a cambio de introducir una sobrecarga considerable, a veces inaceptable, sobre todo en redes de altas prestaciones.

Ninguno de los estudios sobre QoS citados aborda el problema de mantener la QoS ofrecida incluso durante el proceso de reconfiguración. Así, trabajos como FM [21] o BIP [22] no consideran siquiera la posibilidad de cambios en la red.

La aplicación de un protocolo de reconfiguración dinámica puede ser el primer paso para proporcionar garantías de QoS. Al tratarse de redes con pocos recursos no podemos plantearnos usar multitud de canales virtuales o reales, sino que debemos optimizar el uso que se hace de los que se disponen mediante una adecuada planificación.

En redes con reconfiguración dinámica sí podremos intentar dar garantías de QoS. Así pues, algunos de los métodos ya aplicados a otro tipo de redes, van a proporcionar un marco adecuado para tráfico con necesidades de QoS sobre redes con reconfiguración dinámica progresiva. En concreto, una multiplexación que permita la asignación dinámica y distribuida de slots de tiempo para usar los canales de la red, podría proporcionar buenos resultados.

VI. CONCLUSIONES

Los resultados presentados confirman que la técnica de reconfiguración dinámica con evitación de bloqueos para encaminamiento distribuido produce una degradación considerablemente menor que la originada por las técnicas de reconfiguración tradicionales. Estos mismos resultados reafirman nuestro interés en trasladar esta técnica de reconfiguración dinámica a los entornos NOW citados en el artículo, en los cuales se utilizan diferentes estrategias de control de bloqueos y encaminamiento de mensajes. Confiamos en que la aplicación de nuestras técnicas permitan garantizar una determinada QoS en NOWs.

REFERENCES

- [1] R. Casado, M.B. Caminero, P. Cuenca, F.J. Quiles, A. Garrido, and J. Duato, "Una herramienta para el análisis de algoritmos de configuración y encaminamiento en redes irregulares," *VIII Jornadas de Paralelismo*, Sept. 1997.
- [2] R. Casado, F.J. Quiles, J.L. Sánchez, and J. Duato, "An efficient protocol for dynamic reconfiguration in irregular networks," Tech. Rep., UCLM, 1998.
- [3] T.L. Rodeheffer and M.D. Schroeder, "Automatic reconfiguration in autonet," Tech. Rep. 77, Systems Research Center of Digital Equipment Corporation, Sept. 1991.
- [4] Myricom Inc., <http://www.myri.com>.
- [5] R. W. Horst, "Thet: A reliable system area network," *IEEE Micro*, Feb. 1995.
- [6] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection networks. An engineering approach*, IEEE Computer Society, 1997.
- [7] R. Casado, F.J. Quiles, J.L. Sánchez, A. Garrido, and J. Duato, "Encaminamiento libre de bloqueos en redes irregulares dinámicamente reconfigurables," *IX Jornadas de Paralelismo*, Sept. 1998.
- [8] R. Casado, F.J. Quiles, J.L. Sánchez, and J. Duato, "Deadlock-free routing in irregular networks with dynamic reconfiguration," in *Procs. of the CANPC'99, USA*, Jan. 1999.
- [9] W.J. Dally and C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. on Computers*, vol. C-36, no. 5, May 1987.
- [10] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, no. 12, pp. 1320-1331, Dec. 1993.
- [11] P. López, J.M. Martínez, and J. Duato, "A very efficient distributed deadlock detection mechanism for wormhole networks," in *Procs. of the 4th Int. Symposium on HPCA*, Feb. 1998.
- [12] K.V. Anjan and T.M. Pinkston, "Disha: A deadlock recovery scheme for fully adaptive routing," in *Procs. of the 9th Int. Parallel Processing Symposium*, Apr. 1995.
- [13] J.M. Martínez, P. López, and J. Duato, "Software-based deadlock recovery for true fully adaptive routing in wormhole networks," in *Procs. of the 1997 Int. Conf. on Parallel Processing*, Aug. 1997.
- [14] A. M. Mainwaring, B. N. Chun, S. Schleimer, and D. S. Wilkerson, "System area network mapping," in *Procs. of the 9th Annual Symposium on Parallel Algorithms and Architectures (SPAA)*, 1997.
- [15] F. Naquin, "Evaluation d'architecture réseaux locaux haut débit: Myrinet, master's thesis," Tech. Rep., LHPC, Laboratoire d'informatique de Besancon, 1997.
- [16] ATM Forum, *ATM Forum traffic management specification. Version 4.0*, May 1995.
- [17] G. Karlsson, "Asynchronous transfer of video," *IEEE communication Magazine*, vol. 24, no. 8, pp. 118-126, August 1996.
- [18] M. Gerla, B. Kannan, B. Kwan, P. Palnati, S. Walton, E. Leonardi, and F. Neri, "Quality of service support in high-speed, wormhole routing networks," in *Int. Conf. on Network Protocols*, Oct. 1996.
- [19] K.H. Connelly, *FM-QoS a Quality of Service messaging substrate for asynchronous local-area networks with hardware-level network feedback*, Ph. D. Thesis. University of Illinois at Urbana-Champaign, 1999.
- [20] J. Duato, S. Yalamanchili, M.B. Caminero, D. Love, and F.J. Quiles, "MMR: A high-performance multimedia router. architecture and design trade-offs," in *Procs. of the 5th Symposium on HPCA*, Jan. 1999.
- [21] M. Lauria, S. Pakin, and A.A. Chien, "Efficient layering for high speed communication: Fast messages 2.x," in *Procs. of the 7th High Performance Distributed Computing Conf. (Chicago, Illinois)*, July 1998, pp. 28-31.
- [22] L. Prylli and B. Tourancheau, "BIP: A new protocol designed for high performance networking on myrinet," *LNCS*, vol. 1388, pp. 472-488, 1998.