Una Nueva Propuesta para Proporcionar QoS en InfiniBand

Francisco J. Alfaro Cortés, José L. Sánchez García y José Duato Marín

Resumen— **InfiniBand (IBA) es un nuevo estándar para comunicación tanto entre ordenadores como entre distintos dispositivos. La arquitectura que define IBA proporciona una serie de mecanismos que, adecuadamente usados, son capaces de proporcionar los niveles de fiabilidad, disponibilidad, prestaciones, escalabilidad y calidad de servicio (QoS) necesarios para las aplicaciones actuales y futuras.**

En este trabajo vamos a presentar una nueva propuesta para realizar el tratamiento de los requisitos demandados por las aplicaciones. Se presentará también un algoritmo para realizar la configuración de la tabla de arbitraje en función de las demandas realizadas por las aplicaciones. Finalmente, se hará una evaluación mediante simulación de las propuestas realizadas.

Palabras clave— **InfiniBand, QoS, Arbitraje.**

I. Introducción

La interconexión a través de un bus común tiene la gran ventaja de su sencillez, y se ha venido utilizando durante muchos años. Sin embargo, este sistema de comunicación no es capaz de proporcionar los niveles de prestaciones, escalabilidad, fiabilidad y disponibilidad necesarios para los sistemas actuales. De hecho, en muchos sistemas el bus de E/S se ha convertido en el cuello del botella que frena el crecimiento en prestaciones [1].

En agosto de 1999 un grupo de las compañías más importantes fundaron la InfiniBand^{SM} Trade Association (IBTA) para desarrollar una nueva arquitectura de comunicación, InfiniBand Arquitecture (IBA), destinada tanto para comunicación entre distintas máquinas, como entre un procesador y sus dispositivos. La primera especificación de IBA fue publicada en octubre de 2000 [2]. Desde entonces más de 200 compañías y Universidades se han unido a esta asociación. Este grupo ha venido trabajando para desarrollar un nuevo estándar, que parece se va a convertir en los próximos años en el nuevo estándar "de facto" para este tipo de comunicación.

IBA proporciona ancho de banda suficiente para que la red de comunicación ya no sea el cuello de botella en las comunicaciones con las aplicaciones actuales. Además, mientras que la mayoría de los productos de interconexión utilizados actualmente intentan conseguir maximizar la productividad y minimizar la latencia, muchas de las actuales aplicaciones necesitan garantía en cuanto a ancho de banda, latencia máxima acotada, retraso entre llegadas, porcentaje máximo de descarte de paquetes, etc. Así pues, la mayoría de los productos actuales son incapaces de satisfacer las necesidades de este tipo de aplicaciones, y será necesario que InfiniBand sea capaz de adaptarse para conseguir satisfacer las demandas de dichas aplicaciones.

IBA incorpora varios mecanismos capaces de proporcionar calidad de servicio (QoS) a los usuarios. Los tres principales son la segregación del tráfico por medio de distintos niveles de servicio, el uso de distintos canales virtuales y el arbitraje de los puertos de salida en función de las necesidades del tráfico. Sin embargo, IBA no especifica cómo deben usarse estos mecanismos para proporcionar QoS, quedando esta tarea a cargo de los implementadores o del administrador de la red.

En los últimos años, nuestro trabajo se ha centrado en estudiar las distintas maneras posibles de configurar los mecanismos de IBA para proporcionar QoS. Además, en [3], [4] se ha propuesto una clasificación de los distintos tráficos posibles, basada en la previamente planteada por Pelissier [5]. Esta clasificación va a permitir segregar los distintos tipos de tráfico y darle a cada uno de ellos un tratamiento distinto para conseguir satisfacer sus necesidades de QoS.

En [3] se ha propuesto una sencilla estrategia para rellenar la tabla de arbitraje que los puertos de salida incorporan tanto en los hosts como en los conmutadores de InfiniBand. Esta estrategia se ha evaluado en [3], [4] con tráfico CBR y en [6] con tráfico VBR. Ya entonces se apuntaba el problema que tiene esa estrategia. No se puede garantizar ningún tipo de QoS a las aplicaciones que usen la tabla de arbitraje de baja prioridad, si las fuentes no respetan el ancho de banda que tienen concedido. En este trabajo se presenta una nueva propuesta para resolver este problema. De esta forma, todas las aplicaciones conseguirán los niveles de QoS que habían demandado.

La estructura de este trabajo es la siguiente: en la Sección II se presenta un resumen de las características de InfiniBand, centrándose la Sección III en los mecanismos que incorpora IBA para proporcionar QoS. En la Sección IV se verá la nueva propuesta para rellenar la tabla de arbitraje. En la Sección V se presentará la evaluación realizada, para terminar en la Sección VI mostrando algunas conclusiones.

II. InfiniBand

La especificación de la Arquitectura de InfiniBand (IBA) describe una System Area Network (SAN) para interconectar ordenadores entre si, así como con

Departamento de Informática, Escuela Politécnica Superior, Universidad de Castilla-La Mancha, 02071 - Albacete (España). {falfaro,jsanchez}@info-ab.uclm.es

Departamento de Informática de Sistemas y Computadores, Universidad Politécnica de Valencia, 46071 - Valencia (España). jduato@gap.upv.es

Este trabajo ha sido parcialmente financiado por la CICYT con el proyecto TIC2000-1151-C07 y por la Junta de Comunidades de Castilla-La Mancha con el proyecto PBC-02-008.

dispositivos de E/S. La arquitectura propuesta es independiente de las plataformas y de los sistemas operativos utilizados.

IBA está basada en un sistema conmutado con enlaces punto a punto de alta velocidad. Una red de InfiniBand está dividida en subredes interconectadas por encaminadores, teniendo cada subred uno o varios conmutadores, nodos host y/o dispositivos de E/S. IBA permite formar cualquier topología.

Los enlaces de IBA son punto a punto y full-dúplex, y pueden ser de cable de cobre, fibra óptica o circuito impreso. La frecuencia de la señal en los enlaces es 2.5 GHz en la versión 1.0, pudiendo permitirse frecuencias mayores en versiones posteriores. Los enlaces físicos pueden usarse en paralelo para conseguir mayores anchos de banda. Actualmente IBA define tres velocidades distintas. La velocidad más baja (llamada 1x) corresponde a un único enlace de 2.5 GHz, y por tanto permite alcanzar 2.5 Gbps. Las otras velocidades permitidas son 10 Gbps (llamada 4x) y 30 Gbps (llamada 12x) que se corresponden con enlaces de 4 y 12 hilos, respectivamente.

Los conmutadores de IBA encaminan los mensajes utilizando las tablas de encaminamiento que se les programan durante la inicialización de la red o tras una modificación de su topología. Las tablas de encaminamiento sólo consideran la dirección de destino, y no se tiene en cuenta ni el buffer ni el puerto de entrada al conmutador.

El número de puertos que puede tener un conmutador no viene en las especificaciones de IBA, quedando a consideración de los implementadores, aunque está limitado a un máximo de 256 puertos. Los conmutadores pueden conectarse en cascada para formar redes grandes. Además, los conmutadores pueden proporcionar encaminamiento multidestino.

El encaminamiento entre subredes diferentes (a través de encaminadores), está basado en un identificador global de 128 bits de longitud, que modela el esquema de direccionamiento de IPv6. Por otra parte, para el direccionamiento de los conmutadores se usan identificadores locales que sólo permiten 48K nodos finales en una única subred. El resto de 16K direcciones están reservadas para el encaminamiento multidestino.

Los mensajes para ser transmitidos se segmentan en paquetes. El tamaño máximo de paquete puede ser 256 bytes, 1KB, 2KB o 4KB. Todos los paquetes contienen dos CRCs independientes, uno que cubre sólo la parte de datos y que no se modifica en todo el trayecto, y otro que cubre los datos que van cambiando en los conmutadores y encaminadores, y que debe recalcularse cada vez.

Los mecanismos de transporte de IBA proporcionan varios tipos de servicio de comunicación entre nodos finales. Estos tipos pueden ser con conexión o sin conexión (datagram) y ambos pueden ser fiables (con asentimiento) o no fiables.

La administración en IBA está basada en agentes y managers. Mientras que los managers son entidades activas, los agentes son entidades pasivas que responden a los mensajes de los managers. Todas las subredes IBA deben tener un manager maestro, que puede residir en un nodo final o en un conmutador, y que será el encargado, entre otras cosas, de descubrir la topología e inicializar la red.

III. Soporte de IBA para QoS

Principalmente IBA tiene tres mecanismos que van a permitir proporcionar QoS: los niveles de servicio, los canales virtuales y el arbitraje de los puertos de salida.

IBA define un máximo de 16 niveles de servicio (SLs), pero no especifica qué características debe tener el tráfico de cada uno de esos SLs. Así pues, va a depender de la implementación o del administrador de la red cómo distribuir los tipos de tráfico existentes entre los SLs. Al permitir diferenciar los distintos tráficos se podrá dar un tratamiento distinto basado en las necesidades que cada uno de ellos pueda tener.

Los puertos, tanto de los conmutadores y encaminadores como de los hosts, incorporan canales virtuales (VLs). Esto permite soportar múltiples canales virtuales en un único enlace físico. Un VL representa un conjunto de buffers de recepción y transmisión, tal y como puede verse en la Figura 1.

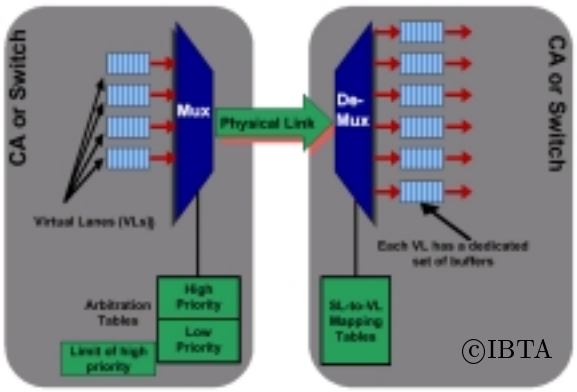

Fig. 1. Estructura de los canales virtuales en un enlace físico.

Los puertos de IBA deben soportar un mínimo de dos y un máximo de 16 canales virtuales (VL_0, ..., VL_{15}). Todos los puertos deben soportar el VL_{15} que está reservado exclusivamente para tareas de administración y debe tener siempre prioridad sobre el resto del tráfico de los otros VLs. Como una red podría construirse con conmutadores con distinto número de canales virtuales, el número de VLs usado por un puerto (que puede ser menor del número real de VLs de que disponga), lo configura el manager de la subred. Además, los paquetes incorporan un SL, y se establece una relación entre SLs y VLs por medio de la tabla SLtoVLMappingTable que especifica en qué VL del conmutador deben situarse los paquetes de un determinado SL.

El control de flujo en cada VL es independiente, de forma que los paquetes de un VL pueden seguir avanzando en su ruta mientras que los de otro VL están detenidos. Cuando un puerto incorpora más de dos VLs, el arbitraje entre los canales virtuales de

datos se va a establecer por un sistema de prioridades en las VLArbitrationTables.

La estructura de una tabla de arbitraje de un puerto de salida es la esbozada en la Figura 2. Cada tabla de arbitraje tiene dos subtablas, una para los paquetes procedentes de los canales virtuales considerados de alta prioridad y otra para los paquetes que están en los canales virtuales considerados de baja prioridad. Sin embargo, IBA no especifica qué es alta y baja prioridad.

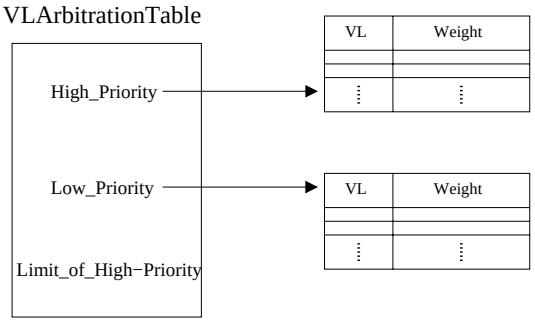

Fig. 2. Estructura de la tabla de arbitraje.

Las tablas de arbitraje funcionan de manera circular con un sistema de pesos. Tanto la tabla de alta como la de baja prioridad pueden tener un máximo de 64 entradas, y cada una de ellas tendría un VL y un peso. Este peso debe ser menor de 255 y especifica el número de unidades de 64 bytes que pueden enviarse por ese VL.

Un valor llamado LimitOfHighPriority especifica el número máximo de paquetes de alta prioridad que pueden enviarse antes de transmitir un paquete de la tabla de baja prioridad. Más concretamente, los VLs de la tabla de alta prioridad pueden transmitir $LimitOfHighPriority \times 4096$ bytes antes que pueda ser transmitido un paquete desde uno de los VLs de la tabla de baja prioridad. Si no hay paquetes que transmitir en ninguno de los VLs de la tabla de alta prioridad, pueden seguir transmitiéndose los paquetes de los VLs de la tabla de baja prioridad.

IV. Rellenado de la Tabla de Arbitraje

En trabajos anteriores hemos propuesto una estrategia para configurar las tablas de arbitraje para tráfico con requisitos de ancho de banda y/o latencia. Esta propuesta presentaba problemas que ya eran apuntados entonces. Comencemos viendo la clasificación de tráfico utilizada.

A. Clasificación del Tráfico

Pelissier propuso una clasificación de tráfico distinguiendo cuatro categorías [5]: DBTS (Dedicated Bandwidth Time Sensitive), DB (Dedicated Bandwidth), BE (Best Effort) y CH (Challenged). En [4], nosotros propusimos extender esa clasificación considerando dos niveles de prioridad para el tráfico BE: PBE (Preferential Best Effort: por ejemplo acceso a webs o bases de datos) y BE (el BE usual como correo, ftp, etc.). Esto permite proporcionar un trato preferente, aunque sin garantías absolutas, a cierto tipo de tráfico BE sobre el resto.

También proponíamos usar la tabla de arbitraje de alta prioridad para el tráfico DBTS, y la tabla de arbitraje de baja prioridad para el resto del tráfico, incluyendo el tráfico DB. Este modelo se usó para evaluar las propuestas realizadas en [3], [4], [6]. Sin embargo, este modelo tiene el problema de no poder proporcionar garantías al tráfico que usa la tabla de baja prioridad si las fuentes que usan la tabla de alta prioridad usan más ancho de banda del que tienen concedido. De esta forma, este modelo no es capaz de proporcionar garantías al tráfico DB, que de acuerdo con este esquema, estaría ubicado en la tabla de baja prioridad.

Para resolver este problema nosotros hemos propuesto que todo el tráfico con necesidad de garantía utilice la tabla de arbitraje de alta prioridad [7], [8]. De esta forma, se estaría usando la tabla de alta prioridad tanto para el tráfico DBTS como para el DB, no habiendo diferencia en el tratamiento recibido por ambas categorías. Sin embargo, esto no es problema pues el tráfico DB puede ser considerado como tráfico DBTS con un deadline lo suficientemente amplio.

En [4] se vio que para garantizar a una conexión una determinada latencia máxima, es necesario que a lo sumo haya una determinada distancia máxima entre dos entradas consecutivas dedicadas al canal virtual que utiliza esa conexión. Esa distancia máxima estará en función de la ruta que siga esa conexión y del límite temporal a conseguir. En [7] se ha propuesto categorizar esas distancias posibles en las siguientes: 2, 4, 8, 16, 32 y 64. Allí se ven detenidamente las razones de ello, y cómo se simplifica todo el tratamiento.

B. Establecimiento de los SLs

Con la nueva propuesta realizada el tráfico se clasifica en distintos SLs en función de la máxima latencia solicitada. En concreto, todas las conexiones pertenecientes al mismo SL necesitarán la misma distancia máxima entre dos entradas consecutivas de la tabla de arbitraje de alta prioridad. De esta forma, si se puede usar un VL para cada SL, todas las conexiones que usen el mismo VL necesitarán la misma máxima distancia entre dos entradas consecutivas de la tabla de arbitraje de alta prioridad.

Con esta estrategia se soluciona el problema de que alguna fuente envíe más información de la que tenía concedida. La tabla de arbitraje cumplirá con la planificación realizada y garantizará al tráfico del resto de VLs los requisitos que solicitaron. Sin embargo, aquellos flujos que compartan VL con la fuente que está teniendo un mal comportamiento sí que se verán afectados. Esto no es posible solucionarlo con el arbitraje pues éste se realiza a nivel de VL y no de flujo.

C. Algoritmo de Llenado de la Tabla

Se presenta ahora el algoritmo para ubicar una nueva petición en la tabla de arbitraje. Para ello

debe seleccionarse una secuencia de entradas de la tabla con una separación máxima determinada entre dos consecutivas de ellas. Debido a la falta de espacio, no se ha incluido aquí la demostración formal, ni las propiedades y teoremas que pueden derivarse de este algoritmo, estando disponibles en [7].

Dada una nueva petición de conexión lo que primero se intenta es utilizar alguna de las secuencias de entradas ya existentes para el mismo VL, si es que existen. Para ello el ancho de banda acumulado por las conexiones que comparten esa secuencia de entradas, más el ancho de banda de la nueva conexión a situar en la tabla, debe ser menor o igual que el máximo acumulable por esa entrada (un peso de 255 por cada entrada de la secuencia). Sólo en el caso de que no sea posible utilizar una secuencia previa será necesario buscar una secuencia nueva en la tabla. En este caso será necesario utilizar el algoritmo que se describe a continuación.

Sea una tabla T, donde la secuencia t_0, t_1, ..., t_{62}, t_{63} son las entradas de la tabla. Cada entrada t_i tiene un peso asociado w_i, que puede variar entre 0 y 255. Se dice que una entrada t_i está *libre* si y sólo si $w_i = 0$.

Dada una tabla T y una distancia $d = 2^i$, se definen los conjuntos $E_{i,j}$ con $i = log_2\ d$ y $0 \leq j < d$, como $E_{i,j} = \left\{ t_{j+n\times 2^i} \quad n = 0, \ldots, \frac{64}{2^i} - 1 \right\}$. Cada conjunto $E_{i,j}$ contiene las entradas de la tabla T separadas por una distancia d, que pueden atender una petición de distancia $d = 2^i$ comenzando por la posición t_j. Se dice que un conjunto $E_{i,j}$ está libre si $\forall t_k \in E_{i,j}$, t_k está libre.

Dada una nueva petición de distancia $d = 2^i$, el algoritmo propuesto inspecciona todos los posibles conjuntos $E_{i,j}$ en un determinado orden, y selecciona el primero de ellos libre. El orden en el que los conjuntos son inspeccionados se basa en la aplicación de la permutación bit-reversal a los valores del intervalo $[0, d-1]$. En concreto, para una nueva petición de distancia máxima $d = 2^i$, el algoritmo selecciona el primer $E_{i,j}$ libre de la secuencia

$$E_{i,R_0}, E_{i,R_1}, \ldots, E_{i,R_{d-1}}$$

donde R_j es la función bit-reversal aplicada a j codificado con i bits.

Por ejemplo, el orden en el que se inspeccionan los conjuntos para una petición de distancia $d = 8 = 2^3$ es $E_{3,0}$, $E_{3,4}$, $E_{3,2}$, $E_{3,6}$, $E_{3,1}$, $E_{3,5}$, $E_{3,3}$ y $E_{3,7}$. Hay que señalar como el algoritmo intenta dejar el mayor hueco posible libre para luego poder atender las peticiones más restrictivas. Por ejemplo, primero se intentan ocupar las entradas pares y luego las impares para que pueda ser posible atender una petición de distancia 2, que requiere la mitad de las entradas.

En [7] se han demostrado una serie de propiedades y teoremas que prueban que este algoritmo es capaz de atender siempre la petición más restrictiva si hay entradas suficientes. Para ello el algoritmo va dejando los huecos libres situados en las posiciones más adecuadas. Otra característica importante de este algoritmo es que no necesita reubicar peticiones ya situadas en la tabla.

Cuando una conexión termina se descuenta su ancho de banda del acumulado por las conexiones que comparten esa secuencia de entradas. Cuando ese ancho de banda acumulado llega a cero, la secuencia puede ser liberada. En ese caso hay que aplicar un algoritmo de desfragmentación para dejar la tabla en condiciones adecuadas para que el algoritmo de llenado funcione de manera correcta. El funcionamiento y propiedades de ese algoritmo de desfragmentación se describen en [7].

V. Evaluación

En esta sección se va a evaluar la propuesta realizada por medio de simulación. Se ha desarrollado un simulador de InfiniBand siguiendo las especificaciones del estándar. Comencemos mostrando el modelo de red y el tipo de tráfico que hemos utilizado.

A. Modelo de Red

Debido a la falta de espacio, aquí se van a mostrar sólo resultados de redes irregulares de 16 conmutadores generadas de forma aleatoria. Usando otras topologías u otros tamaños se obtienen resultados similares.

Los conmutadores son de 8 puertos, teniendo 4 de ellos conectados un host cada uno y los otros 4 puertos se utilizan para interconexión entre conmutadores. Se ha considerado que el conmutador tiene un crossbar no multiplexado, de forma que cada canal virtual tiene su propia entrada/salida de dicho crossbar. También se ha considerado que cada puerto tiene 16 VLs, tanto a la entrada como a la salida del conmutador. Esto permite que cada SL considerado pueda tener su propio VL.

Cada VL tiene su propio buffer tanto a la entrada como a la salida del conmutador. El tamaño del buffer se ha considerado tal que puedan almacenarse cuatro paquetes completos. Se ha probado con dos tamaños de paquete, el mínimo (256 bytes) y el máximo (4096 bytes) que permite IBA, con lo que los tamaños de buffer considerados han sido 1024 y 16384 bytes, respectivamente.

Respecto a la velocidad del enlace se ha probado con las tres velocidades que permite InfiniBand, aunque aquí se van a mostrar sólo los resultados para el caso de 2.5 Gbps. El resto de casos no presentan diferencias con los resultados aquí presentados.

B. Modelo de Tráfico

Para la evaluación de estas propuestas se ha utilizado tráfico CBR generado aleatoriamente. Se han usado 10 SLs para tráfico con necesidades de QoS. Cada uno de ellos tiene requisitos de distancia máxima y/o ancho de banda distintos. Los SLs usados se muestran en la Tabla I. Para las distancias más demandadas (las mayores) se ha hecho una división en función del ancho de banda de las conexiones.

TABLA I

Características de los SLs utilizados.

SL	Distancia máxima	Rango Ancho Banda (Mbps)
0	2	0.064 - 1.55
1	4	0.064 - 1.55
2	8	0.064 - 1.55
3	16	0.064 - 1.55
4	32	0.064 - 1.55
5		1.55 - 64
6	64	0.008 - 0.064
7		0.064 - 1.55
8		1.55 - 64
9		64 - 255

Durante una primera fase de la simulación se intentan establecer conexiones entre un origen y un destino generados aleatoriamente. Cada conexión se asocia aleatoriamente a un SL, y en función de éste se solicita un ancho de banda en el rango correspondiente a ese SL. En función del SL, la conexión también solicita una distancia máxima entre dos entradas consecutivas de la tabla de arbitraje de los conmutadores por los que pasa. Esa solicitud de conexión se estudia en cada conmutador intermedio y si es posible atenderla se le asocia una secuencia de entradas de la tabla ya usadas por otras conexiones (si es posible acumular el ancho de banda sin sobrepasar el máximo), o una que estaba libre.

Cuando no es posible establecer más conexiones se considera que la red ya está cerca de su carga máxima. En ese momento se dejan de intentar conexiones y comienza un periodo transitorio. Tras éste comienza el régimen estacionario donde se toman las medidas que se mostrarán más adelante.

Aunque los resultados obtenidos por el tráfico BE y CH no son el centro de interés de este trabajo, se ha reservado el 20% del ancho de banda disponible para estas categorías que serán atendidas desde la tabla de baja prioridad. De esta forma, las conexiones podrán utilizar el 80% restante de ancho de banda disponible.

C. Resultados de la Simulación

La Tabla II muestra el tráfico inyectado y obtenido de la red (en bytes/ciclo/nodo), la utilización media (en %) y la reserva de ancho de banda (en Mbps) de los interfaces de los hosts y de los puertos de los conmutadores. Hay que insistir en que la máxima utilización alcanzable es 80%, ya que el 20% restante se ha reservado para el tráfico BE y CH.

Hay que señalar que los resultados son muy similares para los dos tamaños de paquete considerados. El tráfico inyectado coincide con el tráfico entregado por la red, con lo que ésta está funcionando de forma correcta y gracias al correcto dimensionado de los recursos disponibles no entra en saturación. El tráfico está próximo al 80% que es el máximo alcanzable. Podría conseguirse una utilización un poco mayor si se lograran establecer más conexiones. Sin embargo, antes de terminar el periodo de establecimiento de conexiones se han realizado muchos reintentos. El problema es que el algoritmo de encaminamiento utilizado (derivado de up*/down*) no hace un buen uso

TABLA II

Tráfico, utilización y reservas obtenidas.

	Tamaño de paquete	
	Pequeño	Grande
T. Inyectado (Bytes/Ciclo/Nodo)	0.7183	0.7011
T. Entregado (Bytes/Ciclo/Nodo)	0.7183	0.7011
Utilización en hosts (%)	71.832	70.123
Utilización en conmutadores (%)	73.096	72.350
Reserva en hosts (Mbps)	1829.699	1822.367
Reserva en conmutadores (Mbps)	1861.969	1860.269

de los enlaces y tiende a utilizar los mismos en un alto porcentaje. De esta forma, hay enlaces más utilizados que otros, y son varios los que están en el 80%. En el caso de la reserva de ancho de banda realizada, también está próxima a los 2 Gbps, que es el máximo alcanzable (el 80% de los 2.5 Gbps del enlace), habiendo varios enlaces en la red en los que se ha reservado el máximo de 2 Gbps.

También se ha medido la distribución del retraso de los paquetes de las conexiones. Para ello se ha calculado el porcentaje de paquetes que llegan antes de unos determinados umbrales. Estos umbrales son diferentes para cada conexión y están en función de su propia latencia máxima (llamada D en las figuras). Los resultados para el tamaño de paquete pequeño, pueden verse en la Figura 3, siendo similares para el tamaño de paquete grande. Podemos observar como en todos los casos los paquetes llegan antes de que cumpla su latencia máxima (D). En el caso de las conexiones de los SLs más restrictivos, los paquetes llegan a su destino con menos margen, pero siempre cumpliendo su requisito de latencia máxima.

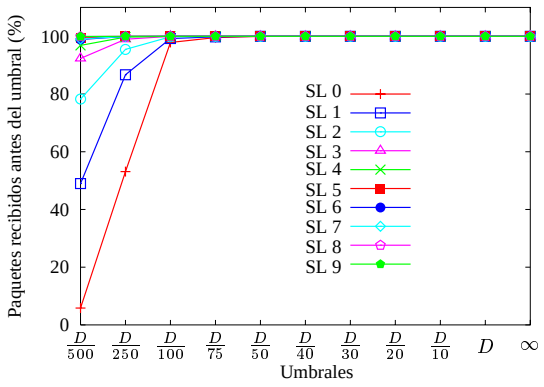

Fig. 3. Distribución del retraso de los paquetes.

Se ha medido también el jitter medio de los paquetes. Se han establecido varios intervalos para medir el porcentaje de paquetes que llegan en cada uno de ellos. Estos umbrales son diferentes para cada conexión y están en función de su propio tiempo entre llegadas (IAT). Los resultados para tamaño de paquete pequeño se muestran en la Figura 4, siendo similares para paquete grande. En todos los casos la mayoría de los paquetes llegan en el intervalo central $[\frac{-IAT}{8}, \frac{IAT}{8}]$. Para los SLs cuyas conexiones tienen ancho de banda medio pequeño (luego su IAT es grande), todos los paquetes llegan en el intervalo central. Para los SLs con mayores anchos de banda (IAT más pequeños), el jitter tiene una distribución Gaussiana.

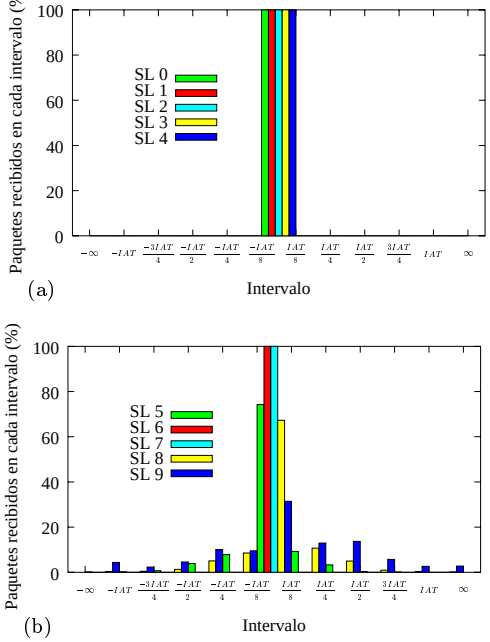

Fig. 4. Jitter medio de los paquetes.

Finalmente, se han seleccionado las conexiones que han entregado mayor y menor porcentaje de paquetes para un determinado umbral. Es decir las conexiones con un mejor y un peor comportamiento para ese umbral. El umbral considerado ha sido $\frac{D}{100}$, donde aún no han llegado todos los paquetes a su destino. Hay que recordar que este umbral es diferente para cada conexión y relativo a su propio requisito en cuanto a latencia máxima. Los resultados para paquete pequeño (para paquete grande son similares) de los SLs 0, 1, 2 y 3, que son los más restrictivos en cuanto a latencia máxima, pueden verse en la Figura 5. Los resultados para el resto de SLs son aún mejores, pues las conexiones de esos SLs son menos restrictivas en cuanto a latencia máxima. Hay que señalar que, incluso las consideradas peores conexiones, obtienen unos resultados que cumplen con creces los requisitos solicitados. También hay que señalar que en todos los casos los resultados obtenidos por la considerada mejor conexión no difieren mucho de los obtenidos por la considerada peor conexión. De esta forma, el arbitraje realizado consigue dar a todas las conexiones un tratamiento similar de forma que todas las conexiones cumplan los requisitos solicitados.

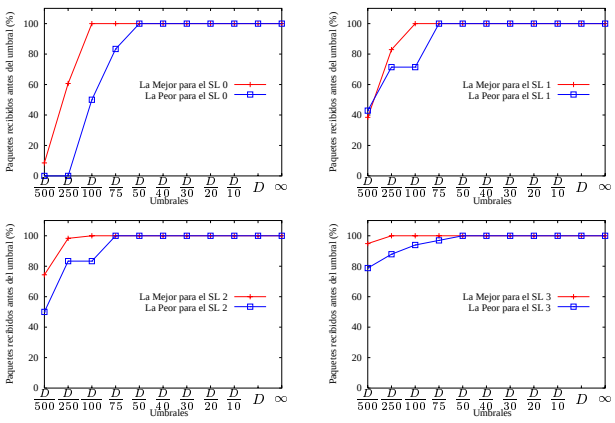

Fig. 5. Comportamiento de la mejor y la peor conexión.

VI. Conclusiones

En este trabajo hemos presentado una breve introducción a InfiniBand centrándonos en los mecanismos que incorpora para proporcionar QoS. En [3] se ha propuesto una nueva metodología para calcular las tablas de arbitraje de InfiniBand proporcionando la QoS solicitada por las conexiones. En [3], [4] se ha evaluado esta propuesta para tráfico con requisitos de ancho de banda y de latencia máxima. Sin embargo, ya entonces se apuntaba que esta metodología plantea problemas si las fuentes no respetan el ancho de banda que previamente solicitaron.

En [7] se ha realizado una nueva propuesta que soluciona los problemas mencionados. Todo el tráfico con necesidades de QoS se trata de forma similar, agrupándose en función de sus requisitos en cuanto a latencia máxima. De esta forma, se ha propuesto que todo el tráfico con necesidad de garantía (DBTS y DB) utilice la tabla de alta prioridad, mientras que la tabla de baja prioridad sólo la utilizarían el tráfico sin necesidades de garantía (PBE, BE y CH).

En [7] se ha propuesto un algoritmo para seleccionar las entradas de la tabla de arbitraje que consigue ubicar una petición si hay entradas suficientes. Para ello, el algoritmo va dejando los huecos en la tabla de la manera más adecuada para poder atender posteriormente a la petición más restrictiva posible.

En la evaluación realizada en [7], [8] y en este trabajo se ha comprobado que la propuesta realizada consigue proporcionar a cada aplicación la QoS que ha demandado durante la fase de establecimiento. También se ha visto que se hace un buen uso de los recursos disponibles, consiguiendo que la red pueda manejar altas tasas de tráfico sin afectar a la QoS recibida por las aplicaciones.

Referencias

[1] G.F. Pfister, *High Performance Mass Storage and Parallel I/O*, chapter 42: An Introduction to the InfiniBand Architecture, pp. 617–632, IEEE Press and Wiley Press, 2001.

[2] InfiniBand Trade Association, *InfiniBand Architecture Specification Volume 1. Release 1.0*, Oct. 2000.

[3] Francisco J. Alfaro, José L. Sánchez, José Duato, and Chita R. Das, "A Strategy to Compute the InfiniBand Arbitration Tables," in *Proceedings of International Parallel and Distributed Processing Symposium (IPDPS'02)*, April 2002.

[4] Francisco J. Alfaro, José L. Sánchez, and José Duato, "A Strategy to Manage Time Sensitive Traffic in InfiniBand," in *Proceedings of Workshop on Communication Architecture for Clusters (CAC'02)*, Apr. 2002, Held in conjunction with IPDPS'02, Fort Lauderdale, Florida.

[5] Joe Pelissier, "Providing Quality of Service over Infiniband Architecture Fabrics," in *Proceedings of the 8th Symposium on Hot Interconnects*, Aug. 2000.

[6] Francisco J. Alfaro, José L. Sánchez, Luis Orozco, and José Duato, "Performance Evaluation of VBR Traffic in InfiniBand," in *Proceedings of IEEE Canadian Conference on Electrical & Computer Engineering (CCECE'02)*, May 2002, pp. 1532 – 1537.

[7] F.J. Alfaro, J.L. Sánchez, M. Menduiña, and J. Duato, "Formalizing the Fill-In of the Infiniband Arbitration Table," Tech. Rep. DIAB-03-02-35, Dep. de Informática Universidad de Castilla-La Mancha, Mar. 2003.

[8] Francisco J. Alfaro, José L. Sánchez, and José Duato, "A New Proposal to Fill in the InfiniBand Arbitration Tables," Submitted to the International Conference on Parallel Computing (ICPP'03), Oct. 2003.