Incrementando las Prestaciones de Ethernet Usando Segment-Based Routing

Andrés Mejía, José Flich, José Duato,
Dept. of Computer Engineering
Universidad Politécnica de Valencia, Spain
{andres, jflich, jduato}@gap.upv.es

Sven-Arne Reinemo y Tor Skeie
Simula Lab
Oslo, Noruega
{svenar,tskeie}@simula.no

Resumen— **En este artículo introducimos un eficiente algoritmo de encaminamiento dentro de la tecnología Ethernet con Control de Flujo 802.3x. Esto hace posible utilizar equipos Ethernet presentes en el mercado con el fín de construir potentes clusters a bajo coste y con un eficiente uso de los componentes de red, lo que contrasta con el inficiente uso de los enlaces que presenta el Spanning Tree protocol (STP). El Segment-Based Routing Algorithm (SR), es un algoritmo de encaminamiento determinista que obtiene elevadas prestaciones sin necesidad de canales virtuales. Además de esto, es agnóstico a la topología, lo que significa que soporta cualquier topología y cualquier combinación de fallos derivada de la topología original. Resultados preliminares muestran como SR es capaz de mejorar las prestaciones de STP, Up*/Down* (UD) y del Tree-based Turn-prohibition.**

Palabras clave— **Encaminamiento, Ethernet, Segment-Based Routing Algorithm.**

I. Introducción

DURANTE los últimos años los clusters de computadores se han convertido en una atractiva solución para la construcción de computadores de altas prestaciones (HPC) y sistemas de servidores. Esta tendencia se ha visto beneficiada por la aparición y difusión de tecnologías que ofrecen una baja latencia y un elevado ancho de banda como son: Myrinet [1], InfiniBand [2], Quadrics [3] y Advanced Switching Interconnect (ASI) [4].

Aunque Ethernet puede ser también usado para construir HPC, esta no ha sido la tecnología predilecta debido a las siguientes razones: Primero, las elevadas latencias experimentadas por los paquetes debido a las diferentes capas de software que deben atravesar (con sus copias asociadas en memoria). Dichas latencias son inapropiadas para HPC. Segundo, los protocolos de reconocimiento y retransmisión requeridos por Ethernet incrementan considerablemente la latencia. Dichos protocolos son necesarios pues Ethernet ha sido diseñada como una red que permite la pérdida de paquetes en presencia de congestión (*lossy networks*). En contraste, tecnologías como Myrinet, InfiniBand y Quadrics que no permiten dichas pérdidas (*lossless networks*), siendo mucho más apropiadas para HPC. Tercero, la falta de un algoritmo de encaminamiento adecuado para HPC ha obligado a los diseñadores de HPC a escoger otras alternativas. Y finalmente, la carencia de un mecanismo de control de congestión en Ethernet hace que sea una opción poco atractiva para HPC, aunque esto está siendo actualmente investigado por el IEEE 802.3ar Congestion Management Task Force[5].

Este trabajo ha sido financiado por el CICYT espaol con la ayuda TIC2003-08154-C06.

Afortunadamente la situación está cambiando y los problemas descritos para construir HPC usando Ethernet se están desvaneciendo lentamente. La introducción de técnicas de Zero Copy Protocols [6] y técnicas de acceso a memoria remotas [7] han ayudado para obtener unas menores latencias. Además, con la introducción en 1997 de mecanismos on/off de control de flujo [8], Ethernet puede comportarse como una red sin pérdidas de paquetes [9] sin la necesidad de costosas retransmisiones y mensajes de reconocimiento. La tecnología Ethernet tiene también algunos beneficios inherentes. El más significativo es su bajo coste, que juega un papel importante en grandes sistemas. Cuando los comparamos con InfiniBand, Myrinet y Quadrics el coste de Ethernet es un orden de magnitud más bajo [10]. Además de esto, Ethernet está más difundida y existen más aplicaciones compatibles con Ethernet, lo que reduce el tiempo de desarrollo para un determinado proyecto. Inclusive, algunas redes de altas prestaciones van tan lejos que hasta incluyen protocolos especiales que mantienen compatibilidad con Ethernet (e.g. Special Protocol Interface en ASI [11]). Pero a pesar de todos estos avances, a día de hoy Ethernet aún carece de un algoritmo de encaminamiento eficiente y adecuado para la construcción de HPC.

A. Algoritmos de Encaminamiento para Ethernet

El encaminamiento en Ethernet es determinista. Esto es, una vez un paquete es inyectado a la red sigue un único camino hasta que alcanza su destino. Uno de los principales beneficios de usar encaminamiento determinista es que preserva el orden de llegada de los paquetes, que es usualmente requerido por las aplicaciones HPC.

El algoritmo de encaminamiento sugerido para Ethernet es el *Spanning Tree Protocol* (STP) [12]. STP basa su funcionamiento en deshabilitar enlaces redundantes de la topología (forzando los enlaces a un estado de bloqueo ó *standby*) con el fín de convertir la tecnología en un árbol [13], lo que puede ser visto como un algoritmo de prohibición de enlaces. Cuando se diseña un sistema HPC es preferible el uso de topologías regulares como mallas y toros. Asimismo es necesario que el algoritmo de encaminamiento garantice que la red es libre de bloqueo, mantenga conectividad entre todos sus nodos y no permita la pérdida de paquetes. Pero el uso de STP es poco apropiado para beneficiarse de las topologías

regulares, debido a que STP forma un árbol a partir de cualquier topología y por lo tanto deshabilita y desperdicia un número considerable de enlaces y elementos de la red.

Muchos algoritmos de encaminamiento han sido propuestos para mejorar las prestaciones del STP, como el SmartBridge [14], STAR [15], y OSR [16]. Sin embargo, todas estas estrategías han sido diseñadas para redes que permiten la pérdida de paquetes sin tener en cuenta los problemas de bloqueo. Otro algoritmo propuesto recientemente, denominado Viking [17], ha tenido un enfoque más ámplio para el reemplazo de STP. Además de intentar diseñar un encaminamiento con rutas mínimas, este esquema también provee equilibrado de la carga de los enlaces y tolerancia a fallos en caso de que ocurriesen fallos de enlaces. Viking explota el concepto de funcionalidad de las Virtual Local Area Network (VLAN) [13] en combinación con Multiples Spanning Trees, dejando que cada etiqueta de VLAN (VLAN tag) decida qué Spanning Tree usar con el fin de encaminar paquetes. A pesar de que este método puede ser implementado sin cambios en el estándar, requiere un gran número de VLANs y un proceso independiente de Viking corriendo en cada terminal de la red. Desafortunadamente, debido a que el estándar Ethernet no permite el uso de buffers dedicados por cada VLAN, Viking puede inducir problemas de bloqueo cuando es usado con mecanismos de control de flujo.

Un algoritmo de encaminamiento determinista apropiado y facilmente implementable para Ethernet con control de flujo es el *Up*/Down** (UD) [18]. UD puede ser utilizado en cualquier topología sin la necesidad de canales virtuales (no soportado por Ethernet), lo que lo hace apropiado para un gran rango de tecnoloigías incluyendo Ethernet [9]. UD calcula un Spanning Tree (después de seleccionar un nodo raíz) y luego según reglas eurísticas coloca restricciones de encaminamiento con el fin de garantizar que la red es libre de bloqueo y mantenga conectividad entre todos sus nodos [19]. Por lo tanto, UD permite el uso de todos sus enlaces al estar basado en restricciones de encaminamiento y no en prohibición de enlaces como el STP. Pero un gran inconveniente que tiene UD es que es sensible a hot-spots alrededor del nodo raíz del spanning tree.

Alguna de las desventajas del UD son explicadas en [20], donde se propone un algoritmo denominado *Tree-based Turn-prohibition* (TBTP). TBTP evita el problema de hot-spot y minimiza el número de restricciones a un máximo de 50% independientemente de la topología usada.

Todos los algoritmos propuestos hasta el momento combinan la libertad de bloqueo con el algoritmo de encaminamiento, pero éste no es el único enfoque posible. En [21] Karol et al. proponen un esquema libre de bloqueo usando una avanzada gestión de buffers. El inconveniente es que éste depende del diametro de la red, cambia la semántica del frame de pausa de Ethernet y requiere una lógica de control adicional para el uso de los buffers.

A pesar de que existen bastantes propuestas de encaminamiento, Ethernet todavía carece de un algoritmo de encaminamiento efectivo que pueda alcanzar el máximo uso posible de los enlaces (sin tener que deshabilitar enlaces) y al mismo tiempo balanceando eficientemente el tráfico. En este artículo, nos formulamos este objetivo proponiendo el uso del Segment-based Routing (SR) en Ethernet. Segment-based routing fue propuesto en [22] y ofrece un novedoso enfoque para posicionar restricciones de encaminamiento de una manera distribuida e independiente. Ésto permite que el algoritmo de encaminamiento explote adecuadamente la topología, incrementando así las prestaciones de la red. SR no requiere de canales virtuales, por lo tanto es compatible con Ethernet.

Asimismo, aplicamos el segment-based routing a redes Ethernet con control de flujo, lo que nos permite construir HPC clusters basados en dispositivos Ethernet. Como primer objetivo, describiremos dos maneras posibles de implementar SR en Ethernet. Como segundo objetivo, comparamos el SR con TBTP y UD, con el fin de exhibir sus prestaciones y escalabilidad. Adicionalmente, y como tercer objetivo mostramos como SR trabaja eficientemente en topologías irregulares.

Adicionalmente, SR puede ser usado en cualquier topología pues permite el descubrimiento de subredes y nodos terminales. En [22] se prueba que SR es capaz de obtener una red libre de bloqueo que mantiene la conectividad entre todos sus nodos sin importar la topología. Esta propiedad (de ser agnóstico a la topología) es obligatoria en grandes redes donde los fallos pueden estar presentes.

El resto del artículo está organizado como sigue, la Sección II provee una visión general del algoritmo SR y como este puede ser usado en ethernet, mientras que en la sección III presentamos los resultados de las simulaciones realizadas y las comparamos con UD y TBTP. En la Sección IV se proponen dos diferentes enfoques de como aplicar el SR en Ethernet y finalmente en la Sección V se presentarán algunas conclusiones y el trabajo futuro.

II. Segment-based Routing

El SR tiene como objetivo primordial obtener las mejores prestaciones de la red mientras mantiene la red libre de bloqueo y se preserva la conectividad entre todos sus nodos. Básicamente, el algoritmo (completamente descrito en [22]) divide (si es necesario) la red en diferentes subredes, y en cada una de estas subredes agrupa los componentes de la red (nodos y enlaces) en segmentos disjuntos, de manera que, una subred esta formada por uno ó mas segmentos y cada componente de la red pertenece a un único segmento en una única subred (i.e. segmentos y subredes son disjuntos). Por ejemplo, la Figura 1(a) nos muestra una topología de nodos (círculos) y enlaces (líneas). A partir de esta topología SR calcula dos subredes diferentes con los segmentos descritos en la Figura 1(b). En particular, 6 segmentos han sido calculados (desde $S1$ hasta $S6$) [1]. Un segmento se define como un conjunto de nodos y enlaces. Así, el segmento $S1$ está formado por los nodos $\{A, B, F, E\}$ y los enlaces $\{1, 2, 3, 4\}$. A su vez, el segmento $S2$ está formado por los nodos $\{C, G\}$ y los enlaces $\{5, 6, 7\}$, mien-

[1]Esta topología ha sido seleccionada con el fin de mostrar todos los casos especiales que presenta el SR.

tras el segmento $S4$ está formado sólo por el enlace $\{11\}$ y así sucesivamente para el resto de segmentos. Nótese que a excepción del segmento inicial, el resto de segmentos, comienza y finaliza en un nodo que pertenece a un segmento préviamente calculado. (e.g. $S2$ comienza/finaliza en $S1$).

Tan pronto como la topología ha sido divida en segmentos, SR añade una restricción bidireccional a cada segmento. Debido a la partición de la red en segmentos independientes, es posible para el SR situar una restricción en un segmento independientemente de los demás segmentos. Cualquier posible combinación finalizará con un algoritmo de encaminamiento que es libre de bloqueo y preserva la conectividad entre todos los nodos de la red (ver demostración en [22]). La Figura 1(b) muestra las posibles restricciones que pueden ser colocadas en cada segmento. Por ejemplo, en $S1$ podemos colocar una restricción bidireccional en los nodos B, E ó F. De todas las posibles combinaciones, SR elige solo una restricción en cada segmento. Para el ejemplo hemos seleccionado una restricción bidireccional en cada uno de los seis segmentos de la Figura 1(c). Nótese que el segmento $S4$ ha recibido un tratamiento especial, puesto que es un segmento unitario. Con el fín de evitar ciclos en la topología es necesario que todo el tráfico que cruce dicho segmento sea evitado. Para lograr esto, seleccionamos uno de los dos nodos adjuntos al segmento unitario y colocamos tantas restricciones como enlaces conectados a segmentos ya restringidos anteriormente. En el ejemplo hemos colocado dos restricciones en el nodo D.

Las restricciones en cada segmento pueden ser colocadas independientemente del lugar donde se han colocado las restricciones en otros segmentos. Esta propiedad dá al SR una gran flexibilidad para elegir el conjunto de restricciones final, lo que mejora las prestaciones de la red. En el ejemplo, existen 144 combinaciones posibles ($3 \times 2 \times 2 \times 2 \times 2 \times 3$) para colocar las restricciones, mientras que para el algoritmo UD existe solo una combinación posible una vez seleccionado el nodo raíz. Ésto sirve como ejemplo para ilustrar la gran flexibilidad que provee SR.

A. Descripción Detallada de SR

En esta sección se describe el algoritmo SR. Para una descripción completa con demostraciones, por favor refiérase a [22]. El SR se basa en tres pasos. Durante el primero se divide la topología en segmentos, después se selecciona una restricción bidireccional para cada segmento y finalmente se selecciona la ruta más adecuada para cada par de nodos.

Con el fín de comenzar con el proceso de segmentación, SR etiqueta cada nodo y enlace con un único identificador. Luego se elige un nodo cualquiera para ser el nodo inicial, este nodo se marca como *visitado* y, a partir de dicho nodo, comienza la busqueda de segmentos en la topología. El algoritmo construye los segmentos visitando nodos y enlaces hasta que se encuentra un nodo ya marcado como *visitado*. Una vez se encuentre un segmento, es etiquetado y todos sus componentes (nodos y enlaces) son marcados como *visitados*. Entonces se busca un nuevo segmento a partir de un nodo *visitado* y con enlaces *no visitados*. Este proceso se repite hasta que no existan mas nodos ni enlaces para visitar.

Durante el proceso de búsqueda de segmentos y dependiendo de la topología, puede ocurrir que no se encuentre ningún nuevo segmento. Por ejemplo, en la Figura 1(b), comenzando desde el nodo H (ya marcado como *visitado*), el algoritmo buscará a través del enlace 17 finalizando en el nodo K sin encontrar un nodo ya *visitado*. En este caso, marcamos el nodo desde el cúal comenzamos la busqueda como *terminal* (T) y el enlace como *bridgelink* (B), lo que significa que ningún segmento puede ser calculado a partir de ellos. Notese que éste es un segmento especial en el que no es necesario colocar una restricción debido a que no puede inducir ningún ciclo en la red. Lo mismo ocurre en el nodo J y enlace 15.

El algoritmo agrupa los segmentos en subredes. Una nueva subred es creada cuando existe al menos un nodo conectado a un *bridgelink* con enlaces no *visitados* conectados a él. En este caso se escoge el nodo conectado al *bridgelink* con enlaces no visitados para ser el nodo inicial de la nueva subred y, a partir de éste, comenzamos la busqueda de nuevos segmentos, hasta que no existan más enlaces o nodos por visitar.

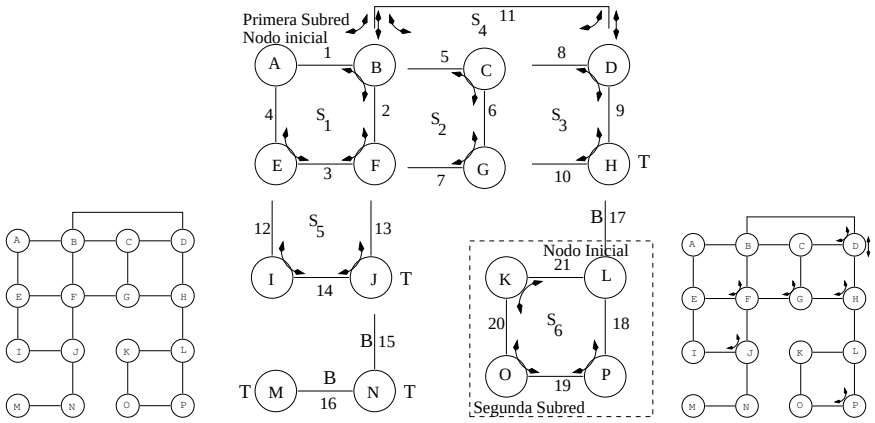
La selección de nodo inicial y el orden en que los enlaces y nodos son buscados influencia el conjunto de segmentos descubiertos, potencialmente influyendo en las prestaciones de la red. En la sección de evaluación proveeremos diferentes criterios que hemos usado para la busqueda de segmentos.

Una vez descubiertos los segmentos, cada uno de éstos puede clasificarse en tres tipos:

- Segmento Inicial. Este segmento (e.g. $S1$ y $S6$ en el ejemplo) comienza y finaliza en el mismo nodo. Este tipo de segmento se encuentra cada vez que una nueva subred es iniciada.
- Segmento Regular. Este tipo de segmento (e.g. $S2$ en el ejemplo) comienza en un enlace, contiene al menos un nodo y finaliza en un enlace.
- Segmento Unitario. Este tipo de segmento (e.g. $S4$ en el ejemplo) contiene sólo un enlace.

Una vez se han obtenido los segmentos, y no existen más elementos de la topología por visitar, el algoritmo SR comienza a situar las restricciones de encaminamiento en cada segmento encontrado. Debido a que la red usa enlaces bidireccionales, debemos situar restricciones bidireccionales. SR coloca las restricciones dependiendo del tipo de segmento. En particular, para un segmento Inicial, se puede ubicar una restriccion bidireccional en cualquier nodo con excepción del nodo inicial (debido a que puede introducir ciclos a través de diferentes subredes). Al hacer esto, eliminamos todos los ciclos en el segmento inicial de la topología. Para segmentos Regulares, se puede ubicar una restricción bidireccional en cualquiera de sus nodos. Esto se realiza con el fín de romper los posibles ciclos que cruzan el segmento. Finalmente, para segmentos Unitarios, se debe evitar que pase tráfico a través de él con el fin de evitar bloqueos. Por lo tanto, en un lado del segmento Unitario se deben colocar tantas restricciones bidireccionales como enlaces conectados a segmentos ya restringidos.

Finalmente, en la tercera parte del algoritmo, se



(a) Topología (b) segmentos, subredes y posibles restricciones (c) Final

Fig. 1. Ejemplo de cálculo del algoritmo SR.

calculan las rutas que seguirán los mensajes teniendo en cuenta las restricciones. La distribución de las rutas seleccionadas influenciará directamente en las prestaciones de la red. La mejor combinación de rutas será aquella que ofrezca la mínima distancia entre cada par de nodos y equilibre el tráfico eficientemente a través de la red. Por otra parte, una selección aleatoria es una solución inmediata para el SR, aunque existen diferentes técnicas que incrementan las prestaciones de la red minimizando el máximo número de rutas que cruzan un enlace, sin embargo ésto tiene un coste significativo.

III. Evaluación de resultados

En esta sección presentamos las simulaciones a nivel de paquetes realizadas en topologías regulares e irregulares. Las simulaciones han sido obtenidas con un simulador Ethernet desarrollado con el J-Sim Framework [23]. Nuestro nodo Ethernet tiene memoria compartida y soporte 802.3x con control de flujo. Cada nodo tiene 5 puertos, uno de ellos está conectado a un nodo terminal y los otros cuatro restantes a otros nodos de la red. Los enlaces Ethernet simulados tienen un ancho de banda de 1 Gbit/s. La tasa promedio de bit es incrementada por pasos desde 10 hasta 1000 Mbit/s (1% - 100% load). El tamaño de los paquetes es de 1522 bytes (máximo tamaño del frame). Los gráficos para visualizar las prestaciones obtenidas son del tipo *Burton Normal Form* [24] (BNF) y usamos dos patrones diferentesde tráfico: uniforme y bit-reversal.

A. Topologías Regulares

Hemos analizado tres conjuntos diferentes de topologías regulares, un toro 8x8, una malla 8x8, y una malla 8x8 con 5% de fallos aleatorios en sus enlaces. Ésta última topología es interesante pues nos permite analizar el potencial del SR cuando se aplica a redes semi-regulares (topologías regulares con fallos en sus enlaces). El cálculo del SR para mallas se ha realizado según el procedimiento descrito en [22].

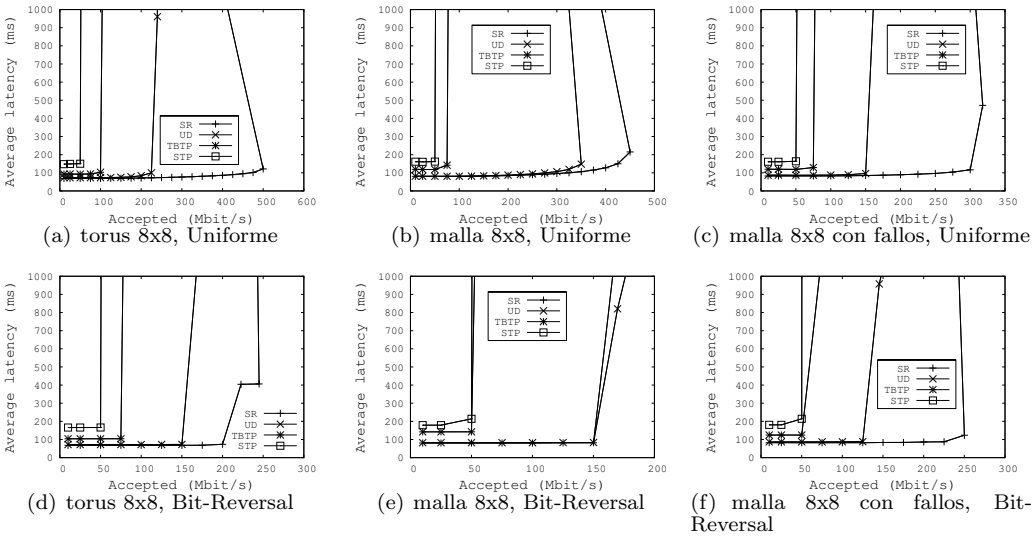
Los resultados para un toro 8x8 mostrados en la Figura 2(a) muestran que SR supera todas las demás alternativas cuando se usa el patrón de tráfico uniforme. UD es el segundo mejor y es superado por un factor de 2.2. UD, TBTP y STP son todos incapaces de balancear el tráfico de forma equilibrada a través de la red como efectivamente lo hace el SR, esto hace que sean incapaces de obtener las mismas prestaciones. Para la malla 8x8 (Figura 2(b)) la tendencia continua. Ahora el SR supera el UD por un factor de 1.2. Aunque la situación ha mejorado para el UD, se ve afectado por una rápida saturación debido al hot-spot asociado en su nodo raíz. Finalmente, para una malla con 5% de fallos en sus enlaces (Figura 2(c)) SR supera UD en un factor de 2. SR retiene gran parte de la regularidad de la malla 8x8 original aún con la presencia de fallos, mientras otras alternativas experimentan una significantiva reducción de sus prestaciones. En general, STP / TBTP son incapaces de explotar las prestaciones de la red puesto que no distribuyen el tráfico de manera eficiente.

Para el patrón de tráfico bit-reversal mostrado en las Figuras 2(d), 2(e) y 2(f), observamos que los resultados son similares, pero con una reducción considerable de las prestaciones debido a que el tráfico inyectado causa un uso no uniforme de los enlaces de la red y por ende una prematura saturación de la misma.

B. Topologias Irregulares

Para topologías irregulares hemos evaluado topologías formadas aleatoriamente con 16, 32 y 64 nodos. Para el cálculo del SR, el nodo con menor ID ha sido seleccionado como nodo inicial, los segmentos han sido calculados buscando la distancia más corta a cualquier nodo ya *visitado*. Por su parte, al situar las restricciones en los segmentos, hemos elegido la posición aleatoriamente y finalmente, las rutas han sido calculadas siguiendo el algoritmo de balanceo de rutas descrito en [25]. Este método minimiza la desviación típica del peso (número de rutas) de los enlaces.

En la Figura 3(a) vemos que el SR incrementa las prestaciones en un factor de 1.5 comparado con UD, mientras que UD y TBTP son muy similares. Para redes de 32 nodos (Figura 3(b)) la diferencia de prestaciones es 1.84 en favor del SR comparado con el UD y de 2.4 en favor del SR cuando es comparado con el TBTP. Por tanto, observamos que a medida que el tamaño de la red aumenta las diferencias en prestaciones también aumentan. TBTP, por



(a) torus 8x8, Uniforme (b) malla 8x8, Uniforme (c) malla 8x8 con fallos, Uniforme

(d) torus 8x8, Bit-Reversal (e) malla 8x8, Bit-Reversal (f) malla 8x8 con fallos, Bit-Reversal

Fig. 2. Productividad para topologías regulares. Tráfico Uniforme and Bit-Reversal.

su parte, sufre más que UD. Para redes con 64 nodos (Figura 3(c)) la tendencia continúa. SR ahora incrementa UD y TBTP en un factor de 2.0 y 3.2 respectivamente.

SR escala mejor que las otras alternativas. Ésto es debido a la ventaja de tener una mejor distribución de tráfico gracias a la flexibilidad y la independencia al colocar las restricciones.

De nuevo, los resultados son similares para un patrón de tráfico Bit-reversal (Figuras. 3(d), 3(e), y 3(f)), pero ahora con menor diferencia debido al uso no uniforme de la red.

Finalmente, indicar que el uso de algoritmos con restricciones de encaminamiento (SR, UD, TBTP) mejora las prestaciones de la red cuando los comparamos con algoritmos de prohibición de enlaces (STP). Ésto es debido a que no se deshabilitan enlaces completamente.

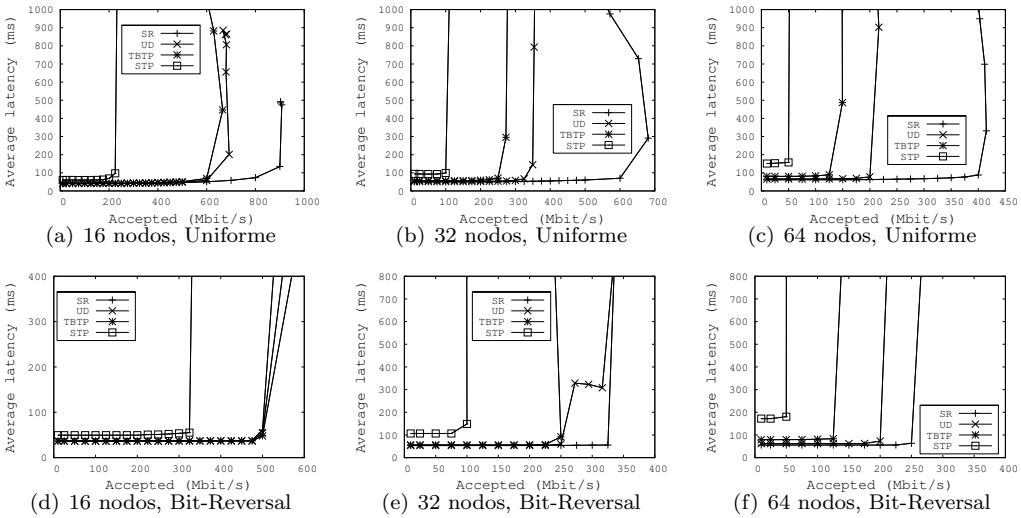
IV. Incorporando SR en Ethernet

El algoritmo STP es el algoritmo de encaminamiento contenido en todos los conmuntadores Ethernet convencionales. Con STP la configuración de equipos Ethernet es una tarea fácil y transparente para el usuario. Cada conmutador ejecuta una instancia del algoritmo para realizar las siguientes operaciones: (a) elegir un *nodo raíz*, (b) negociar la desactivación de los puertos con el fín de construir una red libre de bloqueo (durante este proceso se evitan ciclos asegurándose que sólo un nodo es el responsable del envío de tramas desde la dirección del nodo raíz y hacia un determinado enlace) y (c) escuchar por cambios en la topología (modo de mantenimiento). Mientras la red se encuentra en el modo de mantenimiento todos los nodos están pendientes de un cambio en la topología, y tan pronto como un cambio es detectado es notificado al nodo raíz. Durante dicha fase, el encaminamiento de paquetes puede ser inconsistente.

Con el fín de reemplazar STP por SR hemos considerado dos posibilidades. La primera requiere que los estándares adopten SR como el protocolo de encaminamiento sugerido para ethernet y que el firmware de los futuros conmutadores Ethernet tengan el SR embebido. La segunda posibilidad ofrece una solución menos elegante, pero con algún esfuerzo puede ser utilizada en conmutadores Ethernet actuales.

Claramente, la mejor vía para incorporar SR dentro de Ethernet es reemplazar STP por SR y desarrollar soporte auto-configurable en los mismos niveles que STP lo hace. Ésto requiere la presencia del algoritmo SR en todos los conmutadores y el soporte de las siguientes operaciones: (i) Elegir un nodo maestro. (ii) Recoger información referente a la topología. (iii) Calcular las tablas SR (iv) Distribuir las tablas de encaminamiento. En el paso (i) el nodo maestro se puede seleccionar a través de una negociación como la de la elección del nodo raíz en el paso (a). Después de la elección del nodo maestro, se recoge la información referente a la topología (ii) de los demás nodos y se crea una imagen completa de la red. Posteriormente se calcula (iii) y se distribuyen (iv) las tablas de encaminamiento a todos los nodos. Cuando existe un cambio en la topología, se repiten los pasos (ii), (iii), y (iv).

Nuestra segunda posibilidad no requiere ningún cambio en los conmutadores convencionales Ethernet y se basa en características disponibles en los *managed switches* de Ethernet. Una tabla de encaminamiento para SR requiere que la función de direccionamiento (*route look-up function*) sea capaz de considerar la dirección de destino y el puerto de entrada cuando se selecciona el puerto de salida de la trama que se quiere encaminar. Éste requerimiento es soportado por muchos conmutadores Ethernet clasificados como *managed switches*. Ésto hace posible pre-calcular y distribuir las tablas de encaminamiento de los nodos mediante el uso del *Simple Network Management Protocol (SNMP)*. Las herramientas necesarias requieren ser escritas y los conmutadores tendrán entonces que operar en modo manual (i.e. *STP auto-learning deshabilitado*). Y, en caso de cambios en la topología, nuevas tablas de encaminamiento deben ser cargadas. Claramente, esta posibilidad es un poco molesta al no tener opción de



(a) 16 nodos, Uniforme (b) 32 nodos, Uniforme (c) 64 nodos, Uniforme

(d) 16 nodos, Bit-Reversal (e) 32 nodos, Bit-Reversal (f) 64 nodos, Bit-Reversal

Fig. 3. Productividad para topologías irregulares. Tráfico Uniforme y Bit-Reversal

auto-configuración, aunque las ganancias en productividad son significativas.

V. Conclusiones

Hemos propuesto Segment-based Routing como un algoritmo agnóstico a la topología para Ethernet. Segment-based routing no requiere canales virtuales así que es facilmente implementable en Ethernet. Su novedad radica en la introducción de *independencia local* al ubicar las restricciones de encaminamiento, explotando de esta forma la regularidad de las redes regulares y semi-regulares. Los resultados revelan que SR supera otras alternativas en un factor de 1.2 a 10.0 dependiendo de la topología. Esto indica que SR tiene un gran potencial para convertirse en una metodología de encaminamiento para Ethernet.

Referencias

[1] Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N., Su, W.K.: Myrinet – a gigabit-per-second lan. IEEE MICRO (1995)

[2] InfiniBand Trade Association: Infiniband architecture specification. 1.2 edn. (2004)

[3] Feng, F.P.W., Hoisie., A.: The quadrics network (qsnet): high performance clustering technology. In Sidhu, D.P., ed.: Hot Interconnects, Stanford, California (2001) 125–130

[4] ASI SIG.: Advanced Switching Core Architecture. (2006) http://www.asi-sig.org.

[5] IEEE Standards Association: 802.3ar Congestion Management Task Force. (2006)

[6] Shivam, P., Wyckoff, P., Panda, D.: (emp): Zero-copy (os)-bypass (nic)-driven (gigabit ethernet) message passing. In: ACM/IEEE Conference on High Performance Networking and Computing, Denver, Colorado (2001) 57–57

[7] Nieplocha, V.T.G.S.J., Panda, D.: Host-assisted zero-copy remote memory access communication on infiniband. In: International Parallel and Distributed Computing Symposium, IPDPS04, Santa Fe, NM. (2004)

[8] Seifert, R.: Gigabit Ethernet. Addison Wesley Pub Co. (1998)

[9] Reinemo, S.A., Skeie, T.: Ethernet as a lossless deadlock free system area network. In Pan, Y., Chen, D., Guo, M., Cao, J., Dongarra, J.J., eds.: Parallel and Distributed Processing and Applications: Third International Symposium, ISPA 2005, Nanjing, China, November 2-5, 2005. Proceedings. Volume 3758 of Lecture Notes in Computer Science., Springer Berlin/Heidelberg (2005) 901–914

[10] Dobinson, R.W., Dobson, M., Haas, S., Martin, B., LeVine, M., Saka, F.: Ieee 802.3 ethernet, current status and future prospects at the lhc. In: Nuclear Science Symposium Conference Record, 2000 IEEE. Volume 3. (2000) 26/7–26/13

[11] ASI SIG.: Advanced Switching Core Architecture Specication. (2006) http://www.asi-sig.org/specifications.

[12] IEEE Standards Association: ANSI/IEEE Std 802.1D Media access control (MAC) bridges. (1998)

[13] Seifert, R.: The Switch Book: The Complete Guide to LAN Switching Technology. John Wiley & Sons, Inc. (2000)

[14] Rodeheffer, T.L., Thekkath, C.A., Anderson, D.C.: Smartbridge: a scalable bridge architecture. SIGCOMM Comput. Commun. Rev. **30**(4) (2000) 205–216

[15] Lui, K.S., Lee, W.C., Nahrstedt, K.: Star: a transparent spanning tree bridge protocol with alternate routing. SIGCOMM Comput. Commun. Rev. **32**(3) (2002) 33–46

[16] Garcia, R., Duato, J., Serrano, J.: A new transparent bridge protocol for lan internetworking using topologies with active loops. In: Proceedings of the 1998 International Conference on Parallel Processing, Washington, DC, USA, IEEE Computer Society (1998) 295–303

[17] Sharma, S., Gopalan, K., Nanda, S., Chiueh, T.: Viking: a multi-spanning-tree ethernet architecture for metropolitan area and cluster networks. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Volume 4. (2004) 2283–2294

[18] Schroeder, M.D., Birrell, A.D., Burrows, M., Murray, H., Needham, R.M., Rodeheffer, T.L.: Autonet: a high-speed, self-configuring local area network using point-to-point links. IEEE Journal on Selected Areas in Communications **9**(8) (1991)

[19] Glass, C.J., Ni, L.M.: The turn model for adaptive routing. J. ACM **41**(5) (1994) 874–902

[20] Pellegrini, F.D., Starobinski, D., Karpovsky, M.G., Levitin, L.B.: Scalable cycle-breaking algorithms for gigabit ethernet backbones. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Volume 4. (2004) 2175–2184

[21] Karol, M., Golestani, S.J., Lee, D.: Prevention of deadlocks and livelocks in lossless backpressured packet networks. IEEE/ACM Transactions on Networking **11**(6) (2003) 923–934

[22] Mejia, A., Flich, J., Duato, J., Reinemo, S.A., Skeie, T.: Segment-based routing: An efficient fault-tolerant routing algorithm for meshes and tori. In: Proccedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006), IEEE (2006)

[23] Tyan, H.Y.: Design, Realization, and Evaluation of Component-Based Compsitional Software Architecture for Network Simulation. PhD thesis, Ohio State University (2002)

[24] Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks an Engineering Approach. IEEE Computer Society (1997)

[25] J. Flich, J. Duato, e.a.: Combining in-transit buffers with optimized routing schemes to boost the performance of networks with source routing. In: 2000 International Symposium on High Performance Computing. (2000)