Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Departamento de Sistemas Informáticos

Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas

Trabajo Fin de Máster

Más allá de la tecnología: Topologías toro 3DT.

Septiembre de 2011

Alumno: Francisco José Andújar Muñoz

Directores: José Luis Sánchez García
Francisco José Alfaro Cortés

Firma autor: Firma director/es:

---------- ----------

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1

CURRÍCULUM VÍTAE

Titulación académica

- Ingeniero Técnico en Informática de Sistemas. Escuela Politécnica Superior de Albacete. Universidad de Castilla-La Mancha. Julio de 2008.
- Ingeniero en Informática. Escuela Superior de Ingeniería Informática. Universidad de Castilla-La Mancha. Septiembre de 2010.

Experiencia laboral

- Contratado en proyecto de investigación. Grupo $RAAP$, Instituto de Investigaciones Informáticas de Albacete. Fechas: 06/03/2008-Actualmente.

Participación en proyectos de investigación

- **Título del proyecto**: Técnicas eficientes de encaminamiento y calidad de servicio en redes en chip.
  Entidad financiadora: Junta de Comunidades de Castilla-La Mancha (PCC08-0078-9856).
  Entidades participantes: Universidad de Castilla-La Mancha y Universidad Politécnica de Valencia.
  Duración: 1 de Enero de 2008 - 31 de Diciembre de 2010.
  Investigador responsable: José Luis Sánchez García.
  Número de investigadores participantes: 13.
  Importe total del proyecto: 180.000,00 euros.
- **Título del proyecto**: Mejora de la calidad de servicio ofrecida por la infraestructura de Internet.
  Entidad financiadora: Junta de Comunidades de Castilla-La Mancha (POII10-0289-3724).
  Entidades participantes: Universidad de Castilla-La Mancha.
  Duración: 1 de Abril de 2010 - 31 de Marzo de 2013.

Investigador responsable: Pedro Javier García García.
Número de investigadores participantes: 8.
Importe total del proyecto: 150.000,00 euros.

- **Título del proyecto**: Arquitecturas de servidores, aplicaciones y servicios.
Entidad financiadora: Ministerio de Ciencia e Innovación (TIN2009-14475-C04-03).
Entidades participantes: U. Castilla-La Mancha, U. Murcia, U. Politécnica de Valencia y U. Valencia.
Duración: 1 de Enero de 2010 - 31 de Diciembre de 2012.
Investigador responsable: Francisco José Quiles Flor.
Número de investigadores participantes: 24 en el subproyecto.
Importe total del proyecto: 407.800,00 euros (en el subproyecto)

Publicaciones

Revistas

- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *Optimal Configuration of High-Radix C-switches. Impact on the Network Performance*. IEEE Transactions on Parallel and Distributed Systems (TPDS). Índice de impacto: $1,571$ (JCR 2010). Posición: $65/247$ en el área "Engineering, Electrical & Electronic" del ISI; $27/97$ en el área "Computer Science, Theory & Methods" del ISI. En proceso de revisión.
- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *High-Radix Combined Switches: Formalization and Configuration Methodology*. IEEE Transactions on Computers (TC). En proceso de revisión. Índice de impacto: $1,604$ (JCR 2010). Posición: $63/247$ en el área "Engineering, Electrical & Electronic" del ISI; $12/48$ en el área "Computer Science, Hardware & Architecture" del ISI. En proceso de revisión.

Aportaciones en Congresos

- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *PGAS Model for the Implementation of Scalable Cluster Systems*. First International Workshop on HyperTransport Research and Applications. Mannheim, Alemania, Febrero de 2009.
- F. Triviño García, F. J. Andújar Muñoz, A. Ros Bardisa, J. L. Sánchez García y F. J. Alfaro Cortés. *Sistema Integrado de Simulación de NoCs*. XX Jornadas de Paralelismo (JP2009). A Coruña, España, Septiembre de 2009.

- F. Triviño García, F. J. Andújar Muñoz, A. Ros Bardisa, J. L. Sánchez García y F. J. Alfaro Cortés. *Self-Related Traces: An Alternative to Full-System Simulation for NoCs*. The 2011 International Conference on High Performance Computing & Simulation. Estambul, Turquía, Julio de 2011.

- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *Evaluation of an Alternative for Increasing Switch Radix*. The 10th IEEE International Symposium on Network Computing and Applications (IEEE NCA11). Cambridge, MA, USA, Agosto de 2011.

- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *C-switches: Increasing Switch Radix with Current Integration Scale*. The 13rd IEEE International Conference on High Performance Computing and Communications (HPCC-2011). Banff, Canada, Septiembre de 2011.

- J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín. *Evaluación de una alternativa para aumentar el número de puertos de los conmutadores*. XXII Jornadas de Paralelismo (JP2011). San Cristóbal de la Laguna, España, Septiembre de 2011.

- S. González, F. Triviño García, F. J. Andújar Muñoz, J. L. Sánchez García y F. J. Alfaro Cortés. *Acelerando las simulaciones de sistema completo usando Simics en sistemas multiprocesador*. XXII Jornadas de Paralelismo (JP2011). San Cristóbal de la Laguna, España, Septiembre de 2011.

Technical Report

- DIAB-11-02-1: *An Alternative for Building High-Radix Switches: Formalization and Configuration Methodology.* J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín.
  Departamento de Sistemas informáticos, ESII, UCLM. Febrero 2011.
  `http://www.dsi.uclm.es/trep.php?&codtrep=DIAB-11-02-1`

- DIAB-11-02-2: *An Alternative for Building High-Radix Switches: Application for Special Traffic Patterns.* J. A. Villar Ortiz, F. J. Andújar Muñoz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín.
  Departamento de Sistemas informáticos, ESII, UCLM. Febrero 2011.
  `http://www.dsi.uclm.es/trep.php?&codtrep=DIAB-11-02-2`

- DIAB-11-02-2: *Building 3D torus using low-profile expansion cards* F. J. Andújar Muñoz, J. A. Villar Ortiz, F. J. Alfaro Cortés, J. L. Sánchez García, J. Duato Marín.
  Departamento de Sistemas informáticos, ESII, UCLM. Febrero 2011.
  `http://www.dsi.uclm.es/trep.php?&codtrep=DIAB-11-02-3`

Experiencia en organización de actividades I+D

- Presente y futuro de los sistemas de computación. XX Escuela de Verano de Informática. 21, 22 y 23 de Junio de 2010. Albacete, Vicerrectorado de Extensión Universitaria de la UCLM.

CAPÍTULO 2

RESUMEN DE LAS ASIGNATURAS CURSADAS EN EL MÁSTER

Tecnologías de red de altas prestaciones

En esta asignatura se aporta una visión general de los aspectos más importantes referentes a las redes de interconexión, profundizando en las redes de altas prestaciones.

La primera parte de la asignatura trata la evolución que han seguido las arquitecturas paralelas, para continuar introduciendo al alumno en los principales conceptos necesarios para el diseño de una red de interconexión, como pueden ser la topología, técnicas de conmutación, control de flujo, encaminamiento, etc.

La segunda parte de la asignatura se ha dedicado a la gestión de recursos en Internet, profundizando en servicios integrados, servicios diferenciados y $MPLS$; así como la gestión de recursos en entornos $GRID$ mediante meta-planificadores.

Por último, el Dr. D. José Duato Marín, profesor de la Universidad Politécnica de Valencia, expuso una visión de los futuros servidores de Internet y las $NOCs$ que usarán estos servidores, centrándose en la gestión de la heterogeneidad de estos sistemas, así como el rol que tendrá Hypertransport en el desarrollo de las arquitecturas de sistemas del futuro.

Como trabajo final de la asignatura se presentó la topología toro $3DT$, que recrea un toro de tres dimensiones usando tarjetas de cuatro puertos y parte del estudio teórico sobre sus posibles configuraciones, ambos explicados en este trabajo (capítulo 5).

Modelado y evaluación de sistemas

El principal objetivo de esta asignatura es dar a conocer al alumno los pasos para modelar y evaluar las prestaciones de sistemas informáticos en redes de computadores. Para ello, se estudió el concepto general de sistema y las diversas técnicas que pueden ser utilizadas para su modelado.

El primer bloque de la asignatura se centró en los modelos de simulación, los cuales nos permiten comprender, predecir y controlar el comportamiento del sistema al que representan, profundizando en temas como los diversos modelos de simulación, en qué problemas es útil su aplicación o las ventajas e inconvenientes de simular.

En segundo bloque fue dedicado a la “teoría de colas”, así como su aplicación en modelado y evaluación de sistemas. Para finalizar la asignatura, se realizaron varias prácticas con herramientas de simulación en el ámbito de la investigación en redes de interconexión. Concretamente, fueron utilizados los simuladores Opnet y Network Simulator (NS-2).

En el trabajo final de la asignatura, se presentaron los conmutadores T [27, 28]. Un conmutador T es un conmutador con un alto número de puertos formado por dos conmutadores internos de menor tamaño. Son similares a los nodos virtuales en la topología toro $3DT$, pero aplicado a redes multietapa. En el trabajo se presentó la metodología para configurarlos de forma óptima, además de un estudio teórico sobre la productividad que pueden llegar a alcanzar.

Generación de documentos científicos en Informática

El principal objetivo de esta asignatura es iniciar al alumno en las prácticas más habituales y recomendables del mundo científico e investigador, tales como la generación de documentos de calidad, realizar una correcta documentación de nuestro trabajo, el uso de técnicas estadísticas para poder comparar de forma rigurosa nuestras propuestas con otras existentes, etc.

En el primer bloque se ofrece una introducción al actual sistema de doctorado, criterios de evaluación, baremos, estructura y tramitación de la tesis doctoral. También se describe la estructura habitual de un documento científico de calidad, además de explicar como acceder tanto a las principales fuentes de información en informática como a los principales medios de divulgación científica.

Por otra parte, el segundo bloque se centra en el uso de diferentes técnicas estadísticas para el tratamiento de datos y contraste de distintas hipótesis, además de introducir al alumno en el uso de diferentes herramientas de software estadístico, tales como R y el paquete *R-Commander*.

Para finalizar, en el tercer bloque se realiza una introducción al tratamiento y generación de documentos mediante LaTeX , el cual nos permite la creación de documentos con un aspecto profesional.

El trabajo desarrollado para esta asignatura está destinado a demostrar los conocimientos y capacidades adquiridas. Consiste en la creación de un documento científico en LaTeX cuyo contenido se basa en la aplicación de dos técnicas de inferencia estadística para el contraste de hipótesis.

Las técnicas utilizadas fueron el “t-test” [13, 25] para el test paramétrico y “Kruskal-Wallis” [23] para el test no paramétrico. Concretamente, se usó el “t-test” para comprobar que la productividad alcanzado en una MIN es la misma usando conmutadores T [27, 28] de $k \times k$ puertos y $k/4$ puertos internos que usando conmutadores normales del

mismo tamaño, mientras que el test de “Kruskal-Wallis” fue utilizado para comprobar que la productividad de la red no aumenta si aumentamos el número de enlaces internos del conmutador T más allá del $k/4$.

Introducción a la programación de arquitecturas de altas prestaciones

El principal objetivo de esta asignatura es ofrecer al alumno una visión amplia de la programación en arquitecturas de altas prestaciones, explicando las principales estrategias usadas para mejorar las prestaciones de las aplicaciones científicas, que habitualmente requieren una gran cantidad de cómputo y trabajan con grandes cantidades de información.

Fundamentalmente, la asignatura se centra en dos aspectos: la optimización de código en programas secuenciales mediante la programación orientada a bloques, y la paralelización de código secuencial. Por un lado, la programación orientada a bloques permite obtener mayor rendimiento explotando la jerarquía de memoria de los computadores actuales, mientras que los modelos e ideas básicas sobre la computación paralela permiten al alumno adquirir una metodología de diseño y evaluación de los algoritmos paralelos.

Todos estos conceptos fueron trabajados en las prácticas, donde además se introduce al alumno en la resolución de problemas de álgebra lineal numérica mediante el uso de librerías como $BLAS$ o $PBLAS$ y al uso de librerías de comunicación como MPI o $BLACS$.

Como trabajo de asignatura, se realizó un estudio sobre $TORQUE$ [1], un gestor de colas de código abierto basado en PBS. En el trabajo se exponen las principales características de los gestores de colas y de $TORQUE$, explicando los principales comandos del gestor y haciendo especial hincapié en la creación y lanzamiento de trabajos en clusters administrados con el gestor $TORQUE$. Este trabajo me fue de gran utilidad para el posterior uso que he realizado de los clusters de la universidad, ya que muchos de ellos son administrados mediante $TORQUE$ u otros gestores basados en PBS.

Calidad de Interfaces de Usuario: Desarrollo Avanzado

Esta asignatura se centra en el desarrollo software de interfaces de usuario desde el enfoque de la Ingeniería del Software. Se destacan el papel y la importancia que tienen las $IU's$ en el producto software final, tratando diversas cuestiones como el desarrollo de sistemas interactivos, diseño centrado en el usuario o el proceso de diseño de la interfaz de usuario.

También se trata el desarrollo de interfaces de usuario basado en modelos, ya que la utilización de métodos de diseño con un mayor nivel de abstracción facilita en gran medida la tarea del desarrollo software al promover ciertas características como la portabilidad, interoperabilidad o reusabilidad.

Por último, la asignatura se centra en el concepto de calidad de las interfaces de usuario, haciendo especial hincapié en ciertos requisitos no explícitamente funcionales pero de gran importancia en el desarrollo de $IU's$, como pueden ser la usabilidad, la adaptación o la colaboración.

Como trabajo final de la asignatura se realizó un análisis del modelo de interfaz de usuario concreto en *UsiXml* [2], un lenguaje basado en *XML* que permite la descripción de interfaces de usuario preservando el diseño de forma independiente a las características peculiares de la plataforma de computación física.

Computación en Clusters

En esta asignatura se han estudiado los diferentes aspectos que caracterizan a un cluster de computadores. Un cluster es un tipo de arquitectura distribuida, consistente en un conjunto heterogéneo de computadores conectados a través de una red de área local que se muestran como un único recurso computacional al usuario.

Para ello, se han presentado las últimas tendencias en cuanto a sistemas de interconexión, tales como Infiniband, PCI-X, etc, y de la gestión del espacio del almacenamiento masivo mediante el uso de RAID. Además se han presentado diversas técnicas para mejorar el rendimiento de los sistemas de E/S, ya que en la mayoría de las ocasiones la E/S es el cuello de botella en las aplicaciones ejecutadas en entornos cluster.

Por otra parte, se han analizado los entornos software, conocidos como planificadores, necesarios para poder gestionar todos los recursos del cluster de manera homogénea, así como los diversos entornos software usados para programar aplicaciones paralelas y poder aumentar el rendimiento de nuestras aplicaciones explotando el poder computacional del cluster.

En el trabajo final de la asignatura se presentó el estudio sobre los bloqueos que surgen en la topología toro $3DT$ y las formas de solucionarlo, explicado en el capítulo 6 de este trabajo.

CAPÍTULO 3

INTRODUCCIÓN

En los computadores con un gran número de nodos es habitual el uso de redes de interconexión de altas prestaciones. Puesto que tiene una gran influencia sobre las prestaciones globales del sistema, es esencial un buen diseño de la red de interconexión. Existen múltiples parámetros que deben tenerse en cuenta al realizar el diseño de la red, y existe una fuerte relación entre ellos [12]. Uno de esos parámetros es la topología, la cual determina el patrón de conexión que se acaba formando al unir todos los nodos. Las topologías más utilizadas en los grandes supercomputadores son el fat-tree [21] en el caso de redes indirectas y los toros $3D$ en el caso de las redes directas.

Los toros de tres dimensiones pertenecen a la familia de los n-cubo k-arios [12], que conectan k^n nodos en n dimensiones con k nodos por dimensión. Un toro $3D$ es un 3-cubo k-ario. Este tipo de topologías tienen un grado fijo que facilita la implementación y un diámetro bajo lo cual ayuda a reducir la latencia. Es escalable, siendo lineal el coste de expansión, y admite algoritmos de encaminamiento fáciles de implementar y que pueden ofrecer rutas alternativas para una pareja dada de nodos, lo cual redunda en un alto grado de tolerancia a fallos y la posibilidad de balancear la carga.

Otra característica muy importante de esta topología es que soporta de una forma natural los patrones de comunicación que generan muchas aplicaciones científicas actuales, siendo el ejemplo de aplicación más evidente las simulaciones que modelan espacios de tres dimensiones. El toro $3D$ es la topología usada en varios de los supercomputadores que ocupan las primeras posiciones del top500 [26]. Algunos ejemplos son la familia de los Cray XT (XT4 [16], XT5 [17]) y la familia de los IBM Blue Gene (Blue Gene/L [3], Blue Gene/P [15]).

Para formar un toro $3D$ son necesarios 6 puertos/enlaces en cada nodo, dos por cada dimensión para conectarlo con sus dos vecinos en cada una de ellas. Existen en el mercado tarjetas de comunicación de perfil bajo[1] que disponen de un número reducido de puertos, no siempre suficientes para formar este tipo de topologías. Por ejemplo, con una tarjeta de 4 puertos en cada nodo se podría formar un toro $2D$ pero no uno de tres dimensiones. Sin embargo, si se usan dos tarjetas de este tipo en cada nodo y se emplea un puerto de cada una de ellas para conectarlas entre sí, quedan aún seis puertos libres que podrían utilizarse para conectar al nodo con sus vecinos en un toro $3D$ (figura 5.1, página 24). Al recrear un toro

[1]Una tarjeta de perfil bajo son tarjetas de menos altura que pueden instalarse en los armarios RACK que tienen de altura un U (unidad RACK, $1,75$ pulgadas o $44,45$ mm) para cada máquina.

$3D$ usando las tarjetas de 4 puertos, se puede reducir considerablemente la latencia en la red, utilizando las mismas tarjetas usadas en una red toro $2D$.

Sin embargo, como puede apreciarse en la figura 5.1, es posible que algunas de las rutas que crucen por un nodo lo hagan usando las dos tarjetas, lo cual incrementa el coste de comunicación. Sería conveniente que esa sobrecarga fuera la menor posible, para lo cual se debería conseguir que el mayor volumen de tráfico cruce los nodos utilizando sólo una de las dos tarjetas. Es decir, reducir esa sobrecarga de comunicación pasa por minimizar el tráfico que usa el enlace que conecta las dos tarjetas en cada nodo. Puesto que los seis puertos disponibles por nodo pertenecen a dos grupos de 3 puertos (uno por tarjeta), existen varias formas diferentes de usar esos seis puertos para conseguir el toro $3D$. Y aunque la topología es regular, la sobrecarga de comunicación generada por cada una de esas configuraciones no es la misma, incluso para un tráfico uniforme.

Otro problema de estas configuraciones es que puede llegarse fácilmente a situaciones de bloqueo en la red, también conocidas como *deadlocks*. El problema del *deadlock* surge por el uso del enlace interno, ya que puede ser utilizado por un mensaje independientemente de la dimensión que esté recorriendo en un momento dado. Esto genera nuevos ciclos en la red que no aparecían en el toro $3D$ formado por tarjetas de seis puertos y que no pueden ser eliminados con las técnicas usadas habitualmente en este tipo de redes.

En este trabajo se presenta un estudio detallado del comportamiento de las posibles configuraciones, además de plantear posibles soluciones al problema del *deadlock*. Para empezar, se ha realizado una breve descripción de las características más importantes de las redes de interconexión en el capítulo 4. A continuación se presenta la topología propuesta, a la que se ha denominado toro $3DT$ ($3D$ **T**win), junto con un estudio teórico sobre el comportamiento de las posibles configuraciones del nodo en el capítulo 5. En el capítulo 6 se presenta el algoritmo de encaminamiento utilizado en la red, así como las soluciones obtenidas para eliminar el *deadlock*, para continuar con la evaluación de la topología $3DT$ en el capítulo 7. Para finalizar, se presentan las conclusiones obtenidas del estudio, así como el trabajo futuro, en el capítulo 8.

CAPÍTULO 4

REDES DE INTERCONEXIÓN

En este capítulo se da un breve repaso a diferentes aspectos relacionados con la red de interconexión de los computadores masivamente paralelos. Se describen esencialmente los parámetros de diseño que tradicionalmente han sido usados en el modelado de redes de interconexión, como es el caso de la topología, conmutación, control de flujo y encaminamiento.

4.1. INTRODUCCIÓN

Año tras año se produce un aumento en el rendimiento de los procesadores. El poder incluir en un chip un número cada vez mayor de transistores hace posible incorporar técnicas más complejas y mayor cantidad de memoria, que permiten contribuir a esos elevados niveles de prestaciones. Aunque hay fuertes limitaciones (acceso a memoria, dependencias, retardos en las interconexiones, consumo, etc.) esta tendencia en el rendimiento de los procesadores parece que va a seguir en los próximos años, y todavía con la actual tecnología.

Sin embargo, los computadores con un único procesador no permiten abordar, en tiempos razonables, muchos y muy importantes problemas en las áreas de la ciencia y la ingeniería. Muchos de esos problemas manejan grandes cantidades de datos y todas las operaciones que se realizan con ellos deben completarse en tiempos razonables. La existencia de estos problemas y la necesidad de resolverlos en el menor tiempo posible constituye una de las motivaciones para el uso de sistemas con múltiples procesadores.

Si varios procesadores colaboran para encontrar la solución a un problema se podrá reducir el tiempo necesario para ello. Es necesario que se asignen tareas a los diferentes procesadores disponibles en cada momento, y que éstas se desarrollen con el mayor grado de concurrencia posible. La descomposición del problema en tareas independientes será posible si éste tiene algún tipo de paralelismo implícito. Será el computador el que finalmente aproveche esa característica, siempre y cuando tenga los recursos adecuados para hacerlo.

Hay básicamente dos alternativas para explotar el paralelismo: replicar componentes del sistema y segmentar los componentes de dicho sistema. Los sistemas multiprocesadores obviamente optan por la primera opción, replicando el procesador. El número de procesadores incluidos en los sistemas multiprocesadores varía en función de su uso final. Así, hay ordenadores con un número reducido de elementos de proceso, menos de una decena

de ellos, pero también grandes supercomputadores que llegan a tener centenares de miles de procesadores. En todos los casos es necesario algún tipo de sistema de interconexión que permita una comunicación eficiente entre todos sus componentes.

En general, la red de interconexión está formada básicamente por un conjunto de elementos o interfaces de comunicación y un conjunto de cables que los unen. La información viajará por la red a través de esos elementos, en general pasando por varios de ellos hasta alcanzar su destino. La elección adecuada de los parámetros de diseño de la red de interconexión permitirá obtener un mejor nivel de prestaciones.

4.2. LA RED DE INTERCONEXIÓN

Como se desprende de la sección anterior, la red de interconexión es un componente fundamental en los diferentes modelos de sistemas masivamente paralelos, y en muchos casos es crítica en el nivel de prestaciones que éstos puedan llegar a alcanzar.

Como se ha indicado más arriba, la red de interconexión está formada básicamente por un conjunto de elementos o interfaces de comunicación y un conjunto de cables que los unen. Al patrón de conexión que se acaba formando con todos estos elementos se le denomina *topología*. Establecida la topología, se debe determinar la forma de hacer llegar la información a su destino y a ser posible en el menor tiempo posible. Para ello se han de seleccionar las rutas más adecuadas de entre todas las posibles. De ello se ocupa el mecanismo de *encaminamiento*. Cuando se ha determinado la ruta que la información debe seguir desde un origen a un destino dados, se ha de considerar también la forma en la que dicha información debe avanzar por ella. Será necesario usar una serie de recursos de la red, como canales y buffers, y por tanto se debe establecer la manera en la que dichos recursos son reservados, y posteriormente liberados una vez que dejan de ser necesarios. Hay que decidir cuándo y cómo se produce esa reserva y liberación de recursos, así como qué hacer cuando no es posible llevarlas a cabo. Las técnicas de *conmutación* y *control de flujo* tienen que ver con estas decisiones y son especialmente importantes, sobre todo cuando el grado de utilización de los recursos es elevado.

A la hora de seleccionar la red de interconexión de un sistema dado se tienen en cuenta factores como el rendimiento, la escalabilidad, la fiabilidad y disponibilidad, el coste, el consumo, tamaño del sistema o carga esperada.

4.2.1. Estructura de la red de interconexión

La red de interconexión está formada por conmutadores y enlaces. El conmutador es el elemento de la red que permite que los paquetes se transmitan desde su origen a su destino siguiendo una determinada ruta. Básicamente es un dispositivo con múltiples puertos de entrada y salida. El flujo de datos entra por uno de los puertos y se direcciona hacia

uno o más puertos de salida. Normalmente se usan varios conmutadores y enlaces para construir una red. De esta forma, para que un mensaje llegue de un origen a un destino deberá pasar por uno o más conmutadores. Los conmutadores pueden estar integrados en los nodos de procesamiento, como es el caso de los tradicionales multicomputadores, o constituir elementos independientes de la red, como ocurre en las redes de estaciones de trabajo, o en los clusters.

Los enlaces son los componentes que permiten establecer conexiones físicas punto a punto entre conmutadores (o nodos, si los conmutadores están integrados) o entre conmutador y nodo (en el caso de que el conmutador no esté integrado, en cuyo caso será con la interfaz de red de éste). En muchos casos se usa el término enlace para hacer referencia al medio físico (cable), mientras que se utiliza el término canal para referirse al conjunto que forman el enlace, los controladores de enlace a ambos extremos del mismo, y los correspondientes buffers de almacenamiento. Los enlaces están formados por varios hilos eléctricos, que suelen ser de cobre, o fibras ópticas. La mejor opción, pero también más cara, es la fibra óptica, que consigue mayores anchos de banda y menores tasas de errores, y permite mayores longitudes de cable. Los canales pueden ser unidireccionales o bidireccionales. En el primer caso, la transferencia se produce en un único sentido, mientras que los canales bidireccionales la permiten en ambos. Los canales bidireccionales *full-duplex* permiten esa transmisión en los dos sentidos simultáneamente pues están formados por dos enlaces. Por contra, los canales bidireccionales *half-duplex* sólo están formados por un enlace, y de ahí que en un instante dado sólo se permita la transmisión en uno de los dos sentidos. Los canales se caracterizan por el ancho de banda, que da idea de la velocidad a la que se producen las transferencias. Depende de la frecuencia y del ancho del enlace, y además, otros factores como la distancia, el consumo, el ruido o el tamaño de los buffers influyen sobre el ancho de banda máximo que los canales son capaces de alcanzar.

4.2.2. Clasificación de la redes de interconexión

Según el criterio elegido (modo de funcionamiento, tipo de control, granularidad, etc.) se obtiene una determinada clasificación. Una clasificación ampliamente aceptada, basada principalmente en la estructura de la red, es la que agrupa las redes de interconexión en las siguientes categorías [12]:

a) *Redes de medio compartido*. La red es totalmente compartida por todos los elementos conectados a ella. Las redes de área local y los buses de sistema son los ejemplos más característicos, y el tipo de control, arbitraje, modo de transmisión o asignación y liberación de recursos los aspectos a considerar en su diseño.

b) *Redes directas*. No hay elementos de conmutación en la red, sino que están formando parte de los elementos de procesamiento, los cuales están unidos entre sí mediante conexiones punto a punto. En general, cada uno de ellos está conectado a un subconjunto de los demás, por lo que para hacer llegar mensajes desde uno a otro

cualquiera, en la mayoría de los casos es necesario atravesar varios de ellos. Estas redes son las que se han utilizado en los sistemas multiprocesadores escalables y adoptan estructuras como mallas, toros o hipercubos.

c) *Redes indirectas*. La estructura de la red se configura mediante el uso de elementos de conmutación. Los elementos de proceso, y en su caso la memoria, no se conectan entre sí directamente sino a través de conmutadores. La forma adoptada por la red puede ser regular (crossbar o multietapa) o irregular (red de NOWs y clusters).

d) *Redes híbridas*. Se trata de redes que combinan varios de los tipos anteriores, con el fin de incrementar el ancho de banda de las redes de medio compartido y reducir la latencia de las redes directas o indirectas. Son redes multibus, jerárquicas o basadas en clusters.

4.2.3. Topología

La topología de la red es la representación que se hace de ésta mediante el grafo de interconexión $G(N, C)$, donde N es el conjunto de vértices del grafo, es decir, los nodos[1] de la red, y $C : N \times N$ el conjunto de arcos, formado por los canales físicos que conectan los nodos. La topología determina, a través del grafo G, las relaciones de interconexión existentes entre los diferentes nodos, quedando cada uno de ellos directamente conectado a un subconjunto de nodos de la red a los que se denomina vecinos.

Una topología se evalúa en términos de varios parámetros [12, 10]:

- *Ancho de la bisección* (η). Número mínimo de enlaces que hay que eliminar para dividir la red en dos partes iguales. El ancho de banda de la bisección es la suma del ancho de banda de los enlaces que determinan el ancho de la bisección. Si el tráfico es uniforme la mitad de dicho tráfico atraviesa la bisección.

- *Grado*. El grado de un nodo es el número de conexiones con otros nodos. En el caso de las redes indirectas, el grado se refiere al número de puertos de entrada y salida del conmutador.

- *Diámetro*. Mayor de las distancias mínimas entre todos los posibles pares de nodos.

- *Longitud de los enlaces*. Determina la velocidad a la que la red puede operar y la potencia disipada en ella.

Otros aspectos que también suelen ser considerados son:

[1] Se usa el término nodo para hacer referencia tanto a los nodos de procesamiento, en el caso de redes directas, como a los conmutadores, en el caso de las redes indirectas.

- *Simetría*. Una red es simétrica si desde cualquier nodo la visión que se tiene de ella es la misma. Este tipo de redes hace posible, por ejemplo, que todos los nodos usen el mismo algoritmo de encaminamiento, y simplifica en muchas ocasiones la selección del camino a seguir por los mensajes.

- *Regularidad*. Una red es regular si todos los nodos tienen el mismo grado, el cual puede ser una constante o una función del número de nodos en la red. En un toro de dos dimensiones con canales bidireccionales el grado de cada nodo es 4, mientras que en un hipercubo con n dimensiones el grado de cada nodo es n.

- *Conectividad*. Está relacionada con el número mínimo de enlaces o nodos que deben eliminarse para que la topología se divida en dos o más componentes. Da idea de la robustez de la red, y es considerada como una medida cualitativa de la tolerancia a los fallos en la red.

- *Expansibilidad*. Se refiere a la capacidad de expansión que tiene la red. El número de nodos y enlaces que deben añadirse para conseguirlo varía dependiendo de las características topológicas de la red. Un anillo sólo necesita un nodo para ser expandido, en una malla 2D es preciso insertar una fila y una columna, mientras que en un n-cubo es necesario duplicar el número de nodos.

La topología ideal es aquella que permite mantener una conexión entre cualquier pareja de nodos. Sin embargo, restricciones físicas y económicas la hacen inviable para redes de un cierto tamaño. De ahí que se hayan desarrollado muchas otras alternativas intentando mantener un equilibrio entre prestaciones y coste de desarrollo.

- *Redes directas*. Las topologías para redes directas suelen agruparse atendiendo a su aspecto geométrico. Las más populares y a la vez las que más han sido implementadas son las topologías ortogonales. La topología de una red de interconexión es ortogonal si y sólo si los nodos pueden ser situados en un espacio n-dimensional ortogonal y cualquier conexión se establece de tal forma que produce un desplazamiento en una única dimensión. Los anillos, las mallas, los toros o los hipercubos son topologías ortogonales (figura 4.1a-d).

- *Redes indirectas*. La topología ideal es la que permite disponer de una conexión directa para cualquier pareja de nodos. Esto se traduce en un único conmutador $N \times N$, siendo N el número de nodos en la red (figura 4.1e). Esta red tampoco es viable para redes de gran tamaño. Por ello se han propuesto, y en algunos casos desarrollado, un gran número de topologías en las que se consigue una escalabilidad menos problemática, a costa, claro está, de que un mensaje deba atravesar varios conmutadores, y en varias etapas, para alcanzar su destino. Se puede establecer una clasificación que agrupe a todas esas alternativas atendiendo, por ejemplo, a los aspectos geométricos de las mismas. Así, se habla entonces de topologías regulares y topologías irregulares. En el primer caso se puede distinguir básicamente entre las redes que están formadas a partir

de un único conmutador, o aquellas otras integradas por varios de estos dispositivos, como las multietapas (figura 4.1f).

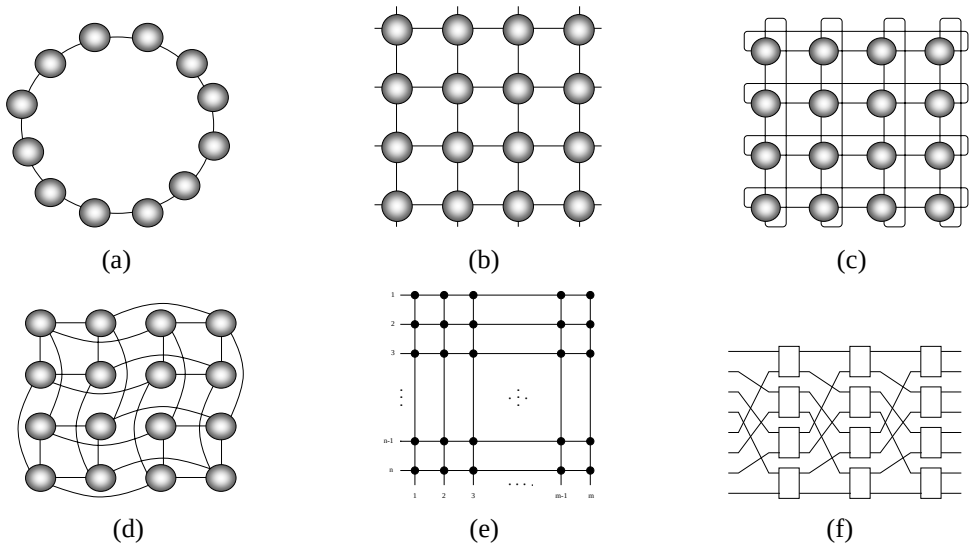


Figura 4.1: Topologías de redes directas: (a) anillo, (b) malla 2D, (c) toro 2D, (d) hipercubo, e indirectas: (e) crossbar, (f) multietapa.

4.2.3.1. Familia n-cubos k-arios

Una red con topología n-cubo k-ario consta de n dimensiones y k nodos por dimensión. A esta familia de topologías pertenecen tanto los anillos, las mallas, los toros y los hipercubos (n-cubos 2-arios). Estas topologías se pueden observar en la figura 4.1. Un anillo no es más que un 1-cubo k-ario, y un toro $2D$ (2-cubo k-ario) se puede construir a partir de k anillos. En general, un n-cubo k-ario se construye a partir de $k(n-1)$-cubos k-arios conectándolos como elementos en k anillos.

Descrito formalmente, un n-cubo k-ario esta formado por $k_0 \times k_1 \times \ldots \times k_{n-2} \times k_{n-1}$ nodos. Cada nodo se identifica mediante un tupla de n elementos, tal que $\langle x_{n-1}, x_{n-1}, \ldots, x_1, x_0 \rangle$, siendo $0 \leq x_i < k_i - 1$ y $0 \leq i \leq n-1$

Estas topologías se caracterizan, para valores pequeños de n, por poseer una ancho de bisección bajo y controlable, grado pequeño, diámetro elevado y enlaces cortos. Existen dos versiones principales de estas redes, dependiendo de si sus extremos están o no unidos, obteniéndose un toro o una malla, respectivamente.

Si el número de dimensiones es pequeño ($n \leq 3$), se puede conseguir enlaces de menor longitud, disminuyendo la disipación de potencia y el retardo del enlace. Es importante ajustar correctamente los valores de k y n: si tenemos mensajes largos, donde el ancho de banda prima sobre la latencia, resulta interesante usar valores grandes de k y

pequeños de n, pero para mensajes cortos, un n grande reduce el diámetro de la red a cambio de ancho de banda. Por ello es interesante ajustar estos valores para poder minimizar la latencia en la red.

4.2.4. Técnicas de conmutación

Las técnicas de conmutación determinan cuándo y cómo se establecen las conexiones internas en los conmutadores entre entradas y salidas, qué mensajes deben utilizar las rutas establecidas, y la forma de asignar los recursos de la red, fundamentalmente enlaces y buffers de almacenamiento.

Los recursos pueden reservarse y liberarse paso a paso o, por el contrario, pueden reservarse al principio todos los recursos que se vayan a necesitar para una vez usados ser liberados, o ser liberados según dejan de usarse. De la política seguida en cada una de estas áreas y la forma de hacerlas interaccionar se obtienen diferentes diseños de los circuitos de comunicaciones que finalmente deben implementarlas.

En general, para transferir un mensaje por la red éste puede dividirse en varios paquetes de tamaño fijo, y éstos a su vez en varios flits, o unidades de control de flujo. Un phit es la unidad de información que puede transmitirse por un canal físico en un ciclo de reloj. Los flits representan unidades de información a nivel lógico, y los phits representan cantidades físicas de bits que pueden transmitirse en paralelo en un ciclo de reloj.

Algunas de las técnicas de conmutación más utilizadas son:

- *Conmutación de circuitos* [7]. A través de un flit de sondeo, que incluye la información de encaminamiento, se establece una ruta física completa entre los nodos fuente y destino para un mensaje dado a base de reservar los recursos necesarios para ello. Establecida la ruta completa, comienza la transferencia de los mensajes y finalizada la recepción del último se procede a la liberación de los canales previamente reservados. Esta técnica es adecuada para transmisión de mensajes largos y en condiciones de poco tráfico en la red. Sin embargo, el coste del establecimiento de la ruta y la baja utilización de los canales son sus principales desventajas.
- *Conmutación de paquetes* [11] El mensaje es dividido en paquetes, cada uno con su propia cabecera con información de encaminamiento. Para comenzar a transmitir un paquete sólo se requiere un canal libre que lo lleve hasta cierto nodo adyacente como etapa intermedia hacia el destino, según la estrategia de encaminamiento empleada. Tiene mayor utilización de los enlaces, pero la latencia tiene una gran dependencia con respecto a la distancia a la que se envía el mensaje, así como con la longitud del mismo.
- *Virtual cut–through*. El paquete sólo es almacenado en un nodo intermedio si, a su llegada a éste, no puede seguir avanzando [20]. El mensaje puede quedar distribuido a

lo largo de los nodos intermedios que formen parte de la ruta entre los nodos fuente y destino, avanzando de forma segmentada. Disminuye la influencia del diámetro de la red sobre la latencia, especialmente cuando los mensajes son largos.

- *Wormhole*. Similar al modelo anterior, se diferencia en que cuando un mensaje llega a un nodo intermedio y no puede avanzar no es almacenado, sino que permanecerá bloqueado hasta que se den las condiciones para reanudar su marcha [8]. Durante el tiempo que transcurre entre ambos eventos, el mensaje mantendrá los recursos que poseía en ese momento, canales y buffers de almacenamiento. Es una técnica de baja latencia, poco sensible a la distancia y que minimiza el coste de almacenamiento de los mensajes. Alcanza niveles de productividad inferiores a los obtenidos con *cut–through*.

4.2.5. Control de flujo

El control de flujo controla el avance de la información entre nodos, es decir, determina el momento en el que la información se transfiere entre componentes del sistema de comunicación (interfaces, buffers, enlaces). Está íntimamente relacionado con la forma de asignar y liberar los buffers que almacenan temporalmente las unidades de información.

El control de flujo se puede realizar a diferentes niveles, por ejemplo extremo a extremo (nodo origen a nodo destino) o a nivel de enlace (de un nodo al siguiente). En cualquier caso se debe establecer un diálogo entre los componentes que deban intervenir. Ese diálogo se puede realizar también de diferentes formas, pero básicamente se trata de enviar señales o mensajes de control para avisar de peticiones y/o para indicar reconocimientos.

Si los enlaces son cortos el control de flujo se implementa con señales de control, mientras que si los enlaces son largos (varios metros de longitud) los mecanismos de control de flujo se basan en el envío de mensajes de control. El control de flujo basado en marcas y el basado en créditos son dos ejemplos bien conocidos [24, 14]:

- *Control de flujo basado en marcas*. La emisión se detiene o se reanuda en base a ciertos niveles de ocupación de los buffers de entrada de los nodos. El más conocido es el protocolo *Stop* & *Go* que está basado en el uso de flits de control y precisa el uso de buffers de entrada más grandes que los habitualmente utilizados en redes con canales no segmentados. Los buffers de entrada están divididos en tres zonas imaginarias separadas por dos marcas: *Stop* y *Go*. Cuando el número de flits en el buffer es tal que se alcanza la marca *Stop*, se envía un flit de control (*Stop*) al nodo desde el cual se están emitiendo los flits. Cuando este nodo recibe dicho flit de control, detiene la emisión de flits en esa dirección. Si por el contrario, se alcanza la marca *Go* según el buffer de entrada se va vaciando, se enviará un flit de control (*Go*) al nodo emisor para que reanude la transmisión de flits.
- *Control de flujo basado en créditos*. Cada nodo recibe inicialmente varios créditos para enviar mensajes a los nodos vecinos. En cada transmisión de un paquete, el emisor

consume un crédito. La emisión se detiene cuando se han consumido todos los créditos. Según se va generando espacio en el receptor, se envían nuevos créditos al emisor.

4.2.6. Encaminamiento

Es el mecanismo que determina el camino que debe seguir un mensaje en la red para alcanzar su destino a partir del nodo fuente en el que se ha generado. Habitualmente son múltiples los caminos que permiten llevar los mensajes a destino, y también varios los de longitud mínima. Una buena estrategia de encaminamiento parece que deba ser la que hace uso precisamente de estos últimos. Hay, además, otros aspectos que suelen tenerse en cuenta a la hora de diseñar algoritmos de encaminamiento, y que unidos a los ya mencionados han dado lugar a complejas y sofisticadas metodologías de diseño de dichos algoritmos.

En general se tiene:

$$R \subset C \times N \times C$$
$$f : \mathcal{S}(C) \times \alpha \longrightarrow C$$

donde la relación R identifica los caminos que puede utilizar un mensaje para alcanzar su destino. Dada la posición actual de un mensaje, C, y su nodo destino, N, la relación R identifica el conjunto de canales, C, que se pueden utilizar para dar el siguiente paso, es decir, alcanzar el siguiente nodo intermedio.

R es una relación, y no una función, ya que puede haber más de un camino posible por donde el mensaje puede continuar. La función f selecciona uno de esos caminos posibles. En cada paso de la ruta, f toma el conjunto de posibles canales $\mathcal{S}(C)$ y cierta información adicional sobre el estado de la red, α, para escoger un canal C por el que continuar. La información α puede ser constante, aleatoria o estar basada en el tráfico de la red.

En el diseño del algoritmo de encaminamiento se debe tener en cuenta no sólo que todos los mensajes alcancen sus correspondientes destinos sino que además lo hagan en el menor tiempo posible. El que esto no ocurra se puede deber a alguno de los siguientes problemas [12]:

- *Deadlock*. Aparece cuando un conjunto de paquetes no puede avanzar por estar a la espera de recursos para poder hacerlo, y esos recursos que necesitan deben ser liberados por paquetes de dicho conjunto (figura 4.2). Si no se evitan, algunos paquetes en la red puede que no lleguen nunca a destino. Las técnicas usadas para el tratamiento de estas situaciones suelen ser básicamente: preventivas, de detección y recuperación, y de evitación.
- *Livelock*. Puede que los paquetes no puedan llegar a su destino sin estar implicados en un *deadlock*. Un mensaje puede estar viajando alrededor del nodo destino sin llegar nunca a alcanzarlo. La mejor solución pasa por usar rutas mínimas.

- *Starvation*. Se da cuando un mensaje permanece parado durante mucho tiempo porque los recursos que solicita son siempre asignados a otros mensajes que también los solicitan. La solución en este caso pasa por usar un esquema correcto de asignación de recursos. Puede ser un esquema basado en una cola circular, o un esquema que, aunque con prioridades, deje parte del ancho de banda para mensajes de baja prioridad.

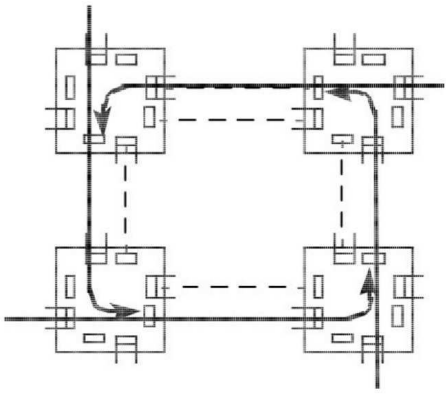


Figura 4.2: Ejemplo de interbloqueo.

Existen múltiples aspectos que caracterizan un algoritmo de encaminamiento [12]. Así por ejemplo, atendiendo al número de destinos a los que va dirigido un mensaje, los algoritmos de encaminamiento se pueden clasificar en *monodestino* y *multidestino*. Dependiendo del lugar donde se tome la decisión de encaminamiento, se habla de encaminamiento *fuente* o encaminamiento *distribuido*. Si se permiten varias rutas para una misma pareja origen-destino se trata de algoritmos *adaptativos* mientras que si sólo existe una ruta posible entre ambos se está ante algoritmos *deterministas*.

4.2.6.1. Encaminamiento y deadlock en n-cubos k-arios

Uno de los algoritmos de encaminamiento más sencillo y ampliamente utilizado en n-cubos k-arios es el algoritmo DOR (**D**imension **O**rder **R**outing). El mecanismo es simple: los mensajes son encaminados por las n dimensiones de la red siguiendo un orden creciente (o decreciente) estricto. Dado que cada nodo indica su posición en la red mediante una tupla $\langle x_{n-1}, x_{n-1}, \ldots, x_1, x_0 \rangle$, los mensajes se envían primero a lo largo de la dimensión 0, después recorren la dimensión 1, y así sucesivamente hasta alcanzar la dimensión $n-1$.

Con el algoritmo DOR, se asegura la ausencia de *deadlock* en hipercubos (n-cubos 2-arios) y mallas n-dimensionales, aunque no ocurre lo mismo con el resto de topologías toroidales, que requieren de otros mecanismos para asegurar la ausencia de bloqueos. Dentro de las soluciones más usadas habitualmente en este tipo de redes, caben destacar el uso de canales virtuales [9] y el control de flujo de la burbuja [6].

Canales Virtuales

La mayoría de los algoritmos de encaminamiento deterministas basan sus propiedades de libertad de bloqueo en la ausencia de bucles en el grafo de dependencia de canales [9].

El grafo de dependencia de canales asociado a una red de interconexión y a un algoritmo de encaminamiento es un grafo dirigido, cuyos vértices representan los canales de la red, mientras que los arcos indican que canales pueden ser usados de forma consecutiva por el algoritmo de encaminamiento. El algoritmo de encaminamiento es libre de bloqueo si y sólo si el grafo de dependencia de canales no tiene bucles [9].

Si el grafo de dependencia contiene bucles, se deben eliminar canales del grafo para eliminar los ciclos. Dado que eliminar canales físicos podría dejar a la red inconexa, se recurre a la multiplexación de los canales (canales virtuales), para posteriormente eliminar aquellos que causan los ciclos en el grafo.

En el caso del algoritmo DOR para redes toroidales, son necesarios al menos dos canales virtuales para poder romper los ciclos. Habitualmente, el conjunto de canales virtuales de divide en dos conjuntos: UP_Links y LOW_Links. Si el destino del mensaje es mayor que el nodo actual, el mensaje se encamina por un canal virtual de UP_Links, en caso contrario se encamina hacia un canal virtual de LOW_Links. Es decir, si el nodo actual es $\langle s_{n-1}, s_{n-1}, \ldots, s_1, s_0 \rangle$ y un mensaje se dirige al nodo $\langle d_{n-1}, d_{n-1}, \ldots, d_1, d_0 \rangle$ a través de la dimensión i, si $d_i > s_i$ se escoge el conjunto de canal virtuales UP_Links, si no se escoge el conjunto LOW_Links. De esta forma, el grafo de dependencia de canales no tiene ciclos y el algoritmo DOR es libre de bloqueo.

Control de flujo de la burbuja

Esta solución elimina el *deadlock* en las topologías n-cubos k-arios a través del control de flujo, por lo que no requiere el uso de canales virtuales [6]. La idea radica en limitar la inyección de paquetes, de forma que no se pueda producir un bloqueo. Para ello, sólo se permite inyectar un paquete en un canal si éste dispone de al menos espacio para dos paquetes, no pudiéndose inyectar si sólo queda espacio para uno nada más, al que se denomina “burbuja”.

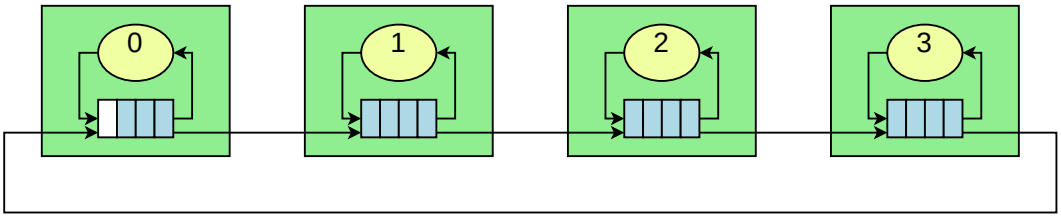


Figura 4.3: Limitación de inyección en un anillo unidireccional.

Supongamos un anillo unidireccional como el de la figura 4.3. Si el nodo 0 inyecta un nuevo paquete en la red, ésta quedará bloqueada. Al permitir la inyección sólo si hay espacio para dos paquetes, el resto de paquetes pueden avanzar en la red, hasta que se genere un segundo hueco y pueda inyectarse un nuevo paquete.

Este mecanismo puede extenderse para toros n dimensionales limitando la inyección entre dimensiones. Si el paquete se desplaza a lo largo de una dimensión, puede avanzar al siguiente conmutador siempre que haya disponible espacio para un paquete. Sin embargo, si el paquete es inyectado desde un nodo o su ruta va a realizar un cambio de dimensión, necesitará dos huecos en el canal de destino para poder seguir avanzando.

CAPÍTULO 5

TOPOLOGÍA TORO 3DT

En este capítulo se presenta el modelo de topología toro $3DT$ (sección 5.1), así como un estudio detallado del comportamiento de las posibles configuraciones de sus nodos para determinar cuál es la mejor de ellas. Puesto que el estudio es dependiente de varios factores, en principio se va a reducir el ámbito de dicho estudio, para lo cual se va a considerar un determinado algoritmo de encaminamiento y patrón de tráfico. A partir de esas condiciones e introducida la notación a utilizar (sección 5.2), en la sección 5.2.1 se presentan las configuraciones a analizar.

Para determinar cuáles de todas las configuraciones posibles ofrecen mejor rendimiento, se determina un procedimiento que es descrito de una manera informal en la sección 5.4.1, para posteriormente abordar el estudio completo de una manera formal en la sección 5.4.2. Previamente, en la sección 5.3 se presenta un estudio más simple considerando únicamente aspectos topológicos, y se compara la topología propuesta, teniendo en cuenta sus posibles configuraciones, con una topología toro $2D$ con el mismo número de elementos de proceso.

5.1. MODELO DE TOPOLOGÍA TORO 3DT

La topología toro $3DT$ no es más que una topología 3-cubo k-ario (toro $3D$) con $k \in \mathbb{N}^*$y $k \geq 2$. La principal diferencia radica en que un nodo en esta topología se puede ver como un *nodo virtual*[1] compuesto por los siguientes componentes principales:

- Hardware de comunicaciones: consistente en dos tarjetas de 4 puertos de tal forma que un puerto de cada una se usa para conectar ambas tarjetas entre sí, y los seis restantes (tres de cada tarjeta) se utilizan para conectar el nodo con sus vecinos.

- Hardware de cálculo: cada tarjeta interna de 4 puertos está conectada a un elemento de proceso, de tal forma que cada nodo posee dos elementos de proceso. Así, la red tiene un total de $2k^3$ elementos de proceso.

[1]Se usa esta expresión para explicar mejor cómo se forman los nodos en esta topología. Sin embargo, de aquí en adelante seguiremos usando sólo *nodo* para referirnos a ellos.

La figura 5.1 muestra un fragmento de la red y el detalle de los nodos de la misma. Como se puede observar, cada nodo contiene dos elementos de proceso ($EP0$ and $EP1$) y dos tarjetas de comunicación ($Card0$ and $Card1$).

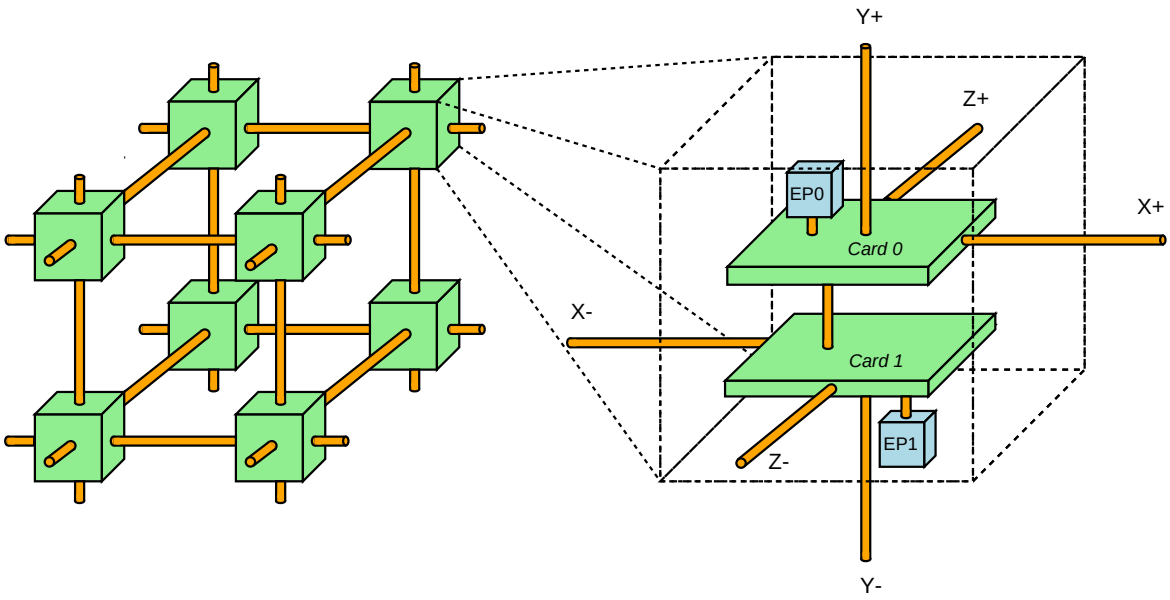


Figura 5.1: Fragmento de un toro $3DT$ y detalle de la circuitería del hardware de comunicaciones, basado en 2 tarjetas de 4 puertos.

5.2. NOTACIÓN

A continuación, se indica la notación utilizada en el estudio. Aunque en algunos casos se simplifica la notación habitualmente usada en casos más generales, ello no supone pérdida de rigurosidad.

- k: número de nodos en cada una de las tres dimensiones del toro $3D$. Se considera inicialmente el mismo número de nodos en cada dimensión.
- $\langle x, y, z \rangle$: identificador de nodo, $0 \leq x, y, z < k$.
- X^-, X^+: puertos/enlaces correspondientes a la dimensión X.
- Y^-, Y^+: puertos/enlaces correspondientes a la dimensión Y.
- Z^-, Z^+: puertos/enlaces correspondientes a la dimensión Z.
- $\mathcal{P}$: conjunto de los puertos de un nodo, $\mathcal{P} = \{X^-, X^+, Y^-, Y^+, Z^-, Z^+\}$.
- P: puerto de un nodo, $P \in \mathcal{P}$.
- $EP0,\ EP1$: elementos de proceso de un nodo.

- $N_s^P(\langle x, y, z\rangle)$: conjunto de nodos que envían mensajes al nodo $\langle x, y, z\rangle$ y llegan a éste por el puerto P.

- $N_d^P(\langle x, y, z\rangle)$: conjunto de nodos a los que el nodo $\langle x, y, z\rangle$ envía mensajes por el puerto P.

- $D_s^P(\langle x, y, z\rangle)$: número de nodos que envían mensajes al nodo $\langle x, y, z\rangle$ y llegan a éste por el puerto P. Es el cardinal de $N_s^P(\langle x, y, z\rangle)$.

- $D_d^P(\langle x, y, z\rangle)$: número de nodos a los que el nodo $\langle x, y, z\rangle$ envía mensajes por el puerto P. Es el cardinal de $N_d^P(\langle x, y, z\rangle)$.

- $R_{P\to P'}(\langle x, y, z\rangle)$: número de rutas que cruzan por el nodo $\langle x, y, z\rangle$ entrando por el puerto P y saliendo por el puerto P'. Cuando P y P' pertenecen a la misma dimensión, a veces se hará referencia a la suma de $R_{P\to P'}(\langle x, y, z\rangle)$ y $R_{P'\to P}(\langle x, y, z\rangle)$ mediante una única expresión, en la que se usará sólo la letra que identifica a la dimensión, omitiendo el signo que identifica la dirección o sentido, y utilizando flecha doble en lugar de simple ($R_{X\leftrightarrow X}(\langle x, y, z\rangle)$, $R_{Y\leftrightarrow Y}(\langle x, y, z\rangle)$ y $R_{Z\leftrightarrow Z}(\langle x, y, z\rangle)$).

- $[a, b]^n$: intervalo que define un conjunto de valores comprendidos entre 0 y $n-1$. La definición 5.1 muestra con más precisión este concepto.

- $D,\ d_{med}$: Diámetro y distancia media de la red, respectivamente.

5.2.1. Configuraciones posibles de la topología

Como los seis puertos disponibles en cada nodo para formar la topología toro $3DT$ pertenecen a dos tarjetas diferentes, es decir a dos grupos distintos de tres puertos, hay varias formas de establecer las conexiones (la figura 5.1 muestra una de ellas).

El número de configuraciones diferentes viene dado por las combinaciones de seis elementos tomados de tres en tres, es decir

$$\binom{6}{3} = \frac{6!}{3!\ 3!} = 20$$

Sin embargo, estos 20 casos se reducen a la mitad puesto que el que un grupo de puertos pertenezca a una tarjeta u otra es indiferente a efectos del estudio que se va a realizar. Las 10 configuraciones distintas consideradas son mostradas en la tabla 5.1, en la que Grupo 1 y Grupo 2 se refieren a los dos grupos de tres puertos correspondientes a cada una de las tarjetas.

Caso	Grupo 1	Grupo 2
A	$\{X^+, Y^+, Z^+\}$	$\{X^-, Y^-, Z^-\}$
B	$\{X^+, Y^+, Z^-\}$	$\{X^-, Y^-, Z^+\}$
C	$\{X^+, Y^+, Y^-\}$	$\{X^-, Z^+, Z^-\}$
D	$\{X^+, Y^+, X^-\}$	$\{Y^-, Z^+, Z^-\}$
E	$\{X^+, Y^-, Z^+\}$	$\{X^-, Y^+, Z^-\}$
F	$\{X^+, Y^-, Z^-\}$	$\{X^-, Y^+, Z^+\}$
G	$\{X^+, Y^-, X^-\}$	$\{Y^+, Z^+, Z^-\}$
H	$\{X^+, Z^+, Z^-\}$	$\{X^-, Y^+, Y^-\}$
I	$\{X^+, Z^+, X^-\}$	$\{Y^+, Y^-, Z^-\}$
J	$\{X^+, Z^-, X^-\}$	$\{Y^+, Y^-, Z^+\}$

Tabla 5.1: Configuraciones posibles de los grupos de puertos de las dos tarjetas.

Algunas de estas configuraciones tienen similar comportamiento, pero sus prestaciones pueden variar en función de las condiciones que sean consideradas a la hora de analizarlas. En las siguientes secciones, se presentan dos estudios con el objetivo de evaluar y comparar el comportamiento de todas las configuraciones incluidas en la tabla 5.1.

En el primer caso, el estudio se realiza sólo desde un punto de vista topológico, para lo cual se consideran dos parámetros característicos de las topologías como son el diámetro y la distancia media. En el segundo estudio se hace intervenir también al algoritmo de encaminamiento y la carga de la red. En este caso, se requiere un análisis mucho más detallado.

5.3. ANÁLISIS DEL TORO 3DT CONSIDERANDO SÓLO LA TOPOLOGÍA

Para tener una primera aproximación de cuál es el comportamiento de las configuraciones, se van a usar parámetros topológicos como el diámetro y la distancia media. Estos parámetros permitirán compararlas. Además, los valores obtenidos servirán también para comparar la topología $3DT$ con una topología $2D$ con el mismo número de elementos de proceso y basada también en tarjetas de 4 puertos. De aquí en adelante, a esa topología $2D$ se le va a denominar *topología toro* $2D$ *equivalente*.

Para realizar este primer estudio comparativo, se van a tener en cuenta las siguientes consideraciones[2]:

[2]Estas consideraciones se tendrán en cuenta sólo para este primer estudio.

- Para facilitar los cálculos, se ha considerado que $k = 2^w$, con $w > 0$. Por lo tanto, la red tendrá un total de $2 \times (2^w)^3 = 2^{3w+1}$ nodos de proceso. Las conclusiones finales del estudio son las mismas si se considera k impar, pero el desarrollo formal es más sencillo si k es par.
- Respecto a la topología toro $2D$ equivalente (mismo número de EPs):
  - Si w es impar, $3w + 1$ es par y es posible construir un toro $2D$ con el mismo número de nodos en cada dimensión. En concreto, la topología equivalente es un toro $2^{\frac{3w+1}{2}} \times 2^{\frac{3w+1}{2}}$.
  - Si w es par, $3w + 1$ es impar y no habrá la misma cantidad de nodos en las dos dimensiones. Así, la topología equivalente que se va a considerar es un toro $2^{\frac{3w}{2}} \times 2^{\frac{3w}{2}+1}$.
- Para calcular el diámetro y la distancia media de los toros $2D$ se usarán las expresiones deducidas a partir de la distancia media y diámetro en los n-cubos k-arios [10].
- El cálculo del diámetro de la topología $3DT$ se explica en detalle en [4].
- La distancia media del toro $3DT$ se ha obtenido mediante simulación, debido a la complejidad de obtener analíticamente esta expresión. El simulador modela una topología toro $3DT$ como la descrita en la sección 5.1. Partiendo de un elemento de proceso origen, se inyecta un mensaje a cada posible destino en la red.

  Un mensaje inyectado se replica cada vez que atraviesa una tarjeta y es enviado por todos los puertos de la misma, excepto por el puerto de recepción. Los mensajes dejan de replicarse al llegar a la tarjeta conectada al EP destino, si la distancia recorrida por el mensaje excede el diámetro de la red, o si la distancia supera el doble de la distancia mínima entre los nodos origen y destino en una red toro $3D$ formada por conmutadores de 6 puertos. De esta forma, se puede obtener la ruta mínima entre cada par de nodos y calcular la distancia media de la red.

En términos de diámetro y distancia media, aparentemente parece que sólo existen dos configuraciones diferentes: una primera configuración en la cual los dos puertos de cada dimensión pertenecen a tarjetas distintas, y una segunda configuración en la cual los dos puertos de una única dimensión están en tarjetas distintas. Las configuraciones A, B, E y F corresponden al primer tipo, mientras que las configuraciones C, D, G, H, I y J corresponden al segundo caso.

Sin embargo, ambos conjuntos de configuraciones ofrecen el mismo diámetro, y la diferencia de distancias medias entre ambos conjuntos es insignificante. Concretamente, los valores obtenidos son:

$$\begin{aligned} D &= 2^{w+1} \\ d_{med} &\approx 2^w \end{aligned}$$

En la tabla 5.2 se incluyen estos resultados y los obtenidos para la topología $2D$ equivalente.

Topología	Dimensiones	D	d_{med}
Toro $3DT$	$2^w \times 2^w \times 2^w$	2^{w+1}	2^w
Toro $2D$ (w par)	$2^{\frac{3w}{2}} \times 2^{\frac{3w}{2}+1}$	$3 \times 2^{\frac{3w}{2}-1}$	$3 \times 2^{\frac{3w}{2}-2}$
Toro $2D$ (w impar)	$2^{\frac{3w+1}{2}} \times 2^{\frac{3w+1}{2}}$	$2^{\frac{3w+1}{2}}$	$2^{\frac{3w+1}{2}-1}$

Tabla 5.2: Diámetro y distancia media de la topología toro $3DT$ y sus equivalentes $2D$.

Comparativa toro 2D / toro 3D

Para finalizar este primer estudio, se comparará los valores obtenidos para la topología toro $3DT$ con su equivalente $2D$. Dado que en todos los casos $D = 2 \times d_{med}$, bastará con usar uno de los dos parámetros para realizar el estudio. En concreto, se ha elegido el diámetro.

- **Si** w **es impar,** se tiene que:

$$\begin{aligned} D_{2D} &> D_{3DT} \\ 2^{\frac{3w+1}{2}} &> 2^{w+1} \\ \frac{3w+1}{2} &> w+1 \\ 3w+1 &> 2w+2 \\ w &> 1 \end{aligned}$$

- **Si** w **es par,** se tiene que:

$$\begin{aligned} D_{2D} &> D_{3DT} \\ 3 \times 2^{\frac{3w}{2}-1} &> 2^{w+1} \\ \log_2 3 + \frac{3w}{2} - 1 &> w+1 \\ \frac{3i}{2} &> w + 2 - \log_2 3 \\ 3w &> 2w + 4 - 2\log_2 3 \\ w &> 4 - 2\log_2 3 \approx 0,830 \end{aligned}$$

Es decir, para $w \geq 2$ presentado ($k \geq 4$, $64\ EPs$ o más), tanto si w es par como si es impar, se tiene una red con menor diámetro y menor distancia media usando la topología toro $3DT$ que usando la topología $2D$ en la que hay una tarjeta por nodo.

5.4. ANÁLISIS DEL TORO 3DT CONSIDERANDO ENCAMINAMIENTO Y TRÁFICO

En este segundo estudio se incorporan el algoritmo de encaminamiento y la carga de la red, con el objetivo de determinar cuál de las diez configuraciones posibles es la configuración óptima bajo estas condiciones. A partir de aquí y en lo que resta del trabajo, se tendrán en cuenta las siguientes hipótesis:

- Se ha considerado un algoritmo de encaminamiento que genera las rutas avanzando en orden creciente de dimensión (DOR, *Dimension* Order Routing [12]). Cuando k es par, para una pareja de nodos origen-destino cualquiera, si la distancia entre ambos es la misma por los dos sentidos de una dimensión dada, la ruta se establece por el enlace que representa en cada dirección el sentido positivo.
- Se ha considerado un patrón de tráfico uniforme. Dadas las características de este tipo de tráfico, los resultados finales del estudio serán los mismos si se considera número de rutas en lugar de número de mensajes. De ahí que a lo largo de este estudio se hablará mayoritariamente en términos de rutas en lugar de mensajes.

5.4.1. Descripción informal

En esta sección se indica de una manera informal cuál será la forma de proceder para obtener la configuración óptima de los puertos de las tarjetas de comunicación en un nodo cualquiera del toro $3DT$. Hay que recordar que la mejor configuración será aquella que minimiza el número de mensajes que cruzan un nodo usando para ello las dos tarjetas, es decir utilizando el enlace que las une.

El número de rutas que cruzan un nodo $\langle x, y, z\rangle$, $0 \leq x, y, z < k$, en general se puede obtener a partir de los posibles nodos origen y nodos destino de las rutas que cruzan por dicho nodo. Así por ejemplo, para calcular las rutas que cruzan el nodo $\langle x, y, z\rangle$ llegando por el puerto X^- y saliendo por el puerto Y^+, se multiplicará el número de nodos que envían mensajes a $\langle x, y, z\rangle$ y que llegan por el puerto X^- por el número de nodos a los que $\langle x, y, z\rangle$ puede enviar mensajes inyectándolos en la red por el puerto Y^+. Esto es

$$R_{X^- \rightarrow Y^+}(\langle x, y, z\rangle) = D_s^{X^-}(\langle x, y, z\rangle) \times D_d^{Y^+}(\langle x, y, z\rangle)$$

Como se mostrará más adelante, esto sólo es cierto cuando el puerto de entrada y el puerto de salida no pertenecen a la misma dimensión. Cuando los dos puertos son de la misma dimensión, ese producto no ofrece el resultado correcto, y éste debe ser obtenido aplicando un proceso algo más elaborado.

En un caso u otro, una vez conocidas las rutas que cruzan por un nodo cualquiera usando una determinada pareja de puertos, con la configuración de las conexiones de las dos tarjetas se puede determinar el número de esas rutas que cruzan por el enlace que las une.

Obtenido este dato para todas las configuraciones se trata finalmente de comprobar cuál es la configuración que minimiza esa cantidad.

Destacar que para calcular el número de rutas, se tendrán en cuenta únicamente los nodos de la red, no los elementos de proceso. Considerando el encaminamiento a nivel de nodo y no a nivel de elemento de proceso, incluir en el estudio la inyección de mensajes a nivel de elemento de proceso multiplicaría por dos los cardinales de los conjuntos de nodos origen y destino de cada puerto P. Esto no afectaría en nada al resultado final, ya que se está multiplicando todas las expresiones por el mismo factor.

Además, tampoco se tendrán en cuenta los mensajes cuyo origen o destino sean el propio nodo. Bajo un tráfico uniforme, cada elemento de proceso del nodo inyectará y recibirá la misma cantidad de mensajes por cada uno de los puertos. Así, o bien los mensajes de $EP0$ tienen que usar el enlace interno para llegar al puerto P, o bien son los mensajes de $EP1$ los que usan el enlace interno para llegar a P. En cualquier caso, la cantidad de mensajes que usan el enlace interno será la misma, sea cual sea la configuración escogida, por lo que mensajes con origen o destino en el propio nodo no afectarán a los resultados que obtengamos para una configuración determinada.

La metodología que se usará para este estudio se puede resumir en los siguientes puntos:

1) Obtener los conjuntos $N_s^P(\langle x, y, z\rangle)$ y $N_d^P(\langle x, y, z\rangle)$, con $0 \leq x, y, z < k$ y $P \in \mathcal{P}$.

2) Obtener los valores $D_s^P(\langle x, y, z\rangle)$ y $D_d^P(\langle x, y, z\rangle)$, con $0 \leq x, y, z < k$ y $P \in \mathcal{P}$.

3) Para las diez configuraciones a estudiar, calcular las rutas $R_{P\to P'}(\langle x, y, z\rangle)$ que cruzan un nodo $\langle x, y, z\rangle$ usando el enlace que une las dos tarjetas internas de dicho nodo, con $0 \leq x, y, z < k$, $P \neq P'$ y $P, P' \in \mathcal{P}$.

4) Determinar la configuración óptima.

Hay que señalar que en todos los cálculos que implican las etapas anteriores, se distinguirá entre k impar y k par puesto que los resultados son distintos para ambos casos.

5.4.2. Descripción formal

Se incluye en esta sección el estudio detallado que se ha realizado para determinar la mejor forma de usar los puertos de las dos tarjetas de comunicación de 4 puertos que permiten en cada nodo donde son incluidas establecer las conexiones con otros nodos vecinos para formar un toro $3DT$. El estudio se presenta siguiendo la metodología que se ha indicado en la sección 5.4.1.

5.4.2.1. Definiciones previas

Incluimos aquí una definición que será útil y simplificará el desarrollo de algunas partes del estudio.

Definición 5.1 *El intervalo $[a,b]^n$, con $b = a + m$; $a, b \in \mathbb{Z}$ y $n, m \in \mathbb{N}$, define el conjunto* $\{(a)\ mod\ n, (a+1)\ mod\ n, \ldots, (a+m-1)\ mod\ n, (a+m)\ mod\ n\}$

Hay que tener en cuenta que al aplicar la operación mod a un número x negativo el resultado que se obtiene es el resto de la división entera entre $x + n$ y n. Así, $(-3)\ mod\ 7 = 4$, o $(-1)\ mod\ 7 = 6$.

Propiedad 5.1 *El número de elementos del conjunto que define el intervalo $[a,b]^n$ es $b-a+1$.*

Demostración: El intervalo $[a,b]^n$ está formado por los valores $a, a+1, a+2, ..., a+m-1, a+m$. Al aplicar la operación mod se obtiene un conjunto con $m+1$ elementos. Como $b = a + m \Rightarrow m = b - a$, y el cardinal del conjunto definido por el intervalo $[a,b]^n$ es $m + 1 = b - a + 1$. □

5.4.3. Conjuntos N_s^P y N_d^P para un nodo $\langle x, y, z\rangle$

A partir de la topología toro $3D$ y el algoritmo de encaminamiento DOR, es fácil determinar los nodos que pertenecen a cada uno de esos conjuntos. Por ello, aquí se indica la composición de los conjuntos N_s^P y N_d^P a través de una serie de definiciones. En cada una de ellas se diferencia entre los casos en los que el número de nodos por dimensión (k) es impar o par. Cuando k es impar, el número de nodos que se pueden alcanzar desde un nodo dado, todos ellos pertenecientes a la misma dimensión, es $\frac{k-1}{2}$, que es el mismo en ambas direcciones. Sin embargo, no ocurre lo mismo en el caso de k par, puesto que hay nodos que se encuentran a la misma distancia topológica en ambas direcciones. Según la decisión que se tome en cuanto a la dirección elegida para alcanzar al nodo, unos enlaces tendrán más carga que otros. Para esos casos, como se establece en las hipótesis iniciales, se ha considerado establecer las rutas por el enlace que representa en cada dirección el sentido positivo.

Definición 5.2 *Sea $N_s^{X^-}(\langle x, y, z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x, y, z\rangle$ y llegan a éste por el puerto X^- ($0 \le x, y, z < k$). Entonces:*

$$N_s^{X^-}(\langle x,y,z\rangle) = \{\ \langle x', y', z'\rangle\ :\ x' \in \begin{cases} \left[x - \frac{k-1}{2}, x-1\right]^k & \textit{si } k \textit{ impar} \\ \left[x - \frac{k}{2}, x-1\right]^k & \textit{si } k \textit{ par} \end{cases},\ y' = y,\ z' = z\}$$

Definición 5.3 *Sea $N_s^{X^+}(\langle x,y,z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x,y,z\rangle$ y llegan a éste por el puerto X^+ ($0 \le x,y,z < k$). Entonces:*

$$N_s^{X^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ :\ x' \in \begin{cases} \left[x+1, x+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[x+1, x+(\frac{k}{2}-1)\right]^k & \textit{si } k \textit{ par} \end{cases},\ y'=y,\ z'=z\}$$

Definición 5.4 *Sea $N_s^{Y^-}(\langle x,y,z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x,y,z\rangle$ y llegan a éste por el puerto Y^- ($0 \le x,y,z < k$). Entonces:*

$$N_s^{Y^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x' < k,\ y' \in \begin{cases} \left[y-\frac{k-1}{2}, y-1\right]^k & \textit{si } k \textit{ impar} \\ \left[y-\frac{k}{2}, y-1\right]^k & \textit{si } k \textit{ par} \end{cases},\ z'=z\}$$

Definición 5.5 *Sea $N_s^{Y^+}(\langle x,y,z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x,y,z\rangle$ y llegan a éste por el puerto Y^+ ($0 \le x,y,z < k$). Entonces:*

$$N_s^{Y^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x' < k,\ y' \in \begin{cases} \left[y+1, y+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[y+1, y+(\frac{k}{2}-1)\right]^k & \textit{si } k \textit{ par} \end{cases},\ z'=z\}$$

Definición 5.6 *Sea $N_s^{Z^-}(\langle x,y,z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x,y,z\rangle$ y llegan a éste por el puerto Z^- ($0 \le x,y,z < k$). Entonces:*

$$N_s^{Z^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x',y' < k,\ z' \in \begin{cases} \left[z-\frac{k-1}{2}, z-1\right]^k & \textit{si } k \textit{ impar} \\ \left[z-\frac{k}{2}, z-1\right]^k & \textit{si } k \textit{ par} \end{cases}\}$$

Definición 5.7 *Sea $N_s^{Z^+}(\langle x,y,z\rangle)$ el conjunto de nodos que envían mensajes al nodo $\langle x,y,z\rangle$ y llegan a éste por el puerto Z^+ ($0 \le x,y,z < k$). Entonces:*

$$N_s^{Z^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x',y' < k, z' \in \begin{cases} \left[z+1, z+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[z+1, z+(\frac{k}{2}-1)\right]^k & \textit{si } k \textit{ par} \end{cases}\}$$

Definición 5.8 *Sea $N_d^{X^-}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto X^- ($0 \le x,y,z < k$). Entonces:*

$$N_d^{X^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ :\ x' \in \begin{cases} \left[x-\frac{k-1}{2}, x-1\right]^k & \textit{si } k \textit{ impar} \\ \left[x-(\frac{k}{2}-1), x-1\right]^k & \textit{si } k \textit{ par} \end{cases},\ 0 \le y',z' < k\}$$

Definición 5.9 *Sea $N_d^{X^+}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto X^+ ($0 \leq x,y,z < k$). Entonces:*

$$N_d^{X^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' \in \begin{cases} \left[x+1, x+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[x+1, x+\frac{k}{2}\right]^k & \textit{si } k \textit{ par} \end{cases},\ 0 \leq y', z' < k\}$$

Definición 5.10 *Sea $N_d^{Y^-}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto Y^- ($0 \leq x,y,z < k$). Entonces:*

$$N_d^{Y^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' = x,\ y' \in \begin{cases} \left[y-\frac{k-1}{2}, y-1\right]^k & \textit{si } k \textit{ impar} \\ \left[y-(\frac{k}{2}-1), y-1\right]^k & \textit{si } k \textit{ par} \end{cases},\ 0 \leq z' < k\}$$

Definición 5.11 *Sea $N_d^{Y^+}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto Y^+ ($0 \leq x,y,z < k$). Entonces:*

$$N_d^{Y^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' = x,\ y' \in \begin{cases} \left[y+1, y+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[y+1, y+\frac{k}{2}\right]^k & \textit{si } k \textit{ par} \end{cases},\ 0 \leq z' < k\}$$

Definición 5.12 *Sea $N_d^{Z^-}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto Z^- ($0 \leq x,y,z < k$). Entonces:*

$$N_d^{Z^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' = x,\ y' = y,\ z' \in \begin{cases} \left[z-\frac{k-1}{2}, z-1\right]^k & \textit{si } k \textit{ impar} \\ \left[z-(\frac{k}{2}-1), z-1\right]^k & \textit{si } k \textit{ par} \end{cases}\ \}$$

Definición 5.13 *Sea $N_d^{Z^+}(\langle x,y,z\rangle)$ el conjunto de nodos a los que el nodo $\langle x,y,z\rangle$ envía mensajes por el puerto Z^+ ($0 \leq x,y,z < k$). Entonces:*

$$N_d^{Z^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' = x,\ y' = y,\ z' \in \begin{cases} \left[z+1, z+\frac{k-1}{2}\right]^k & \textit{si } k \textit{ impar} \\ \left[z+1, z+\frac{k}{2}\right]^k & \textit{si } k \textit{ par} \end{cases}\ \}$$

5.4.4. Valores de D_s^P y D_d^P para un nodo $\langle x,y,z\rangle$

Aplicando la propiedad 5.1 a los conjuntos definidos en la sección 5.4.3 se obtienen los cardinales de dichos conjuntos, esto es los valores de D_s^P y D_d^P para un nodo $\langle x,y,z\rangle$. En la tabla 5.3 se incluyen los valores de D_s^P y en la tabla 5.4 los valores de D_d^P.

$$D_s^{X^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{si } k \text{ impar} \\ \dfrac{k}{2} & \text{si } k \text{ par} \end{cases}$$

$$D_s^{X^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{si } k \text{ impar} \\ \dfrac{k}{2}-1 & \text{si } k \text{ par} \end{cases}$$

$$D_s^{Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{si } k \text{ impar} \\ \dfrac{k^2}{2} & \text{si } k \text{ par} \end{cases}$$

$$D_s^{Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{si } k \text{ impar} \\ \left(\dfrac{k}{2}-1\right)k & \text{si } k \text{ par} \end{cases}$$

$$D_s^{Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{si } k \text{ impar} \\ \dfrac{k^3}{2} & \text{si } k \text{ par} \end{cases}$$

$$D_s^{Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{si } k \text{ impar} \\ \left(\dfrac{k}{2}-1\right)k^2 & \text{si } k \text{ par} \end{cases}$$

Tabla 5.3: Número de nodos que envían mensajes al nodo $\langle x, y, z\rangle$ y llegan a éste por cada uno de sus puertos.

$$D_d^{X^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{si k impar} \\ \left(\dfrac{k}{2}-1\right)k^2 & \text{si k par} \end{cases}$$

$$D_d^{X^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{si k impar} \\ \dfrac{k^3}{2} & \text{si k par} \end{cases}$$

$$D_d^{Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{si k impar} \\ \left(\dfrac{k}{2}-1\right)k & \text{si k par} \end{cases}$$

$$D_d^{Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{si k impar} \\ \dfrac{k^2}{2} & \text{si k par} \end{cases}$$

$$D_d^{Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{si k impar} \\ \dfrac{k}{2}-1 & \text{si k par} \end{cases}$$

$$D_d^{Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{si k impar} \\ \dfrac{k}{2} & \text{si k par} \end{cases}$$

Tabla 5.4: Número de nodos a los que el nodo $\langle x, y, z\rangle$ envía mensajes por cada uno de sus puertos.

5.4.5. Rutas que cruzan por el nodo $\langle x, y, z\rangle$

En esta sección se calcula el número de rutas que cruzan por un nodo $\langle x, y, z\rangle$, distinguiendo por cada pareja de puertos posibles. Para ello, y en la mayoría de los casos, se usarán las expresiones obtenidas en la sección 5.4.4, de tal forma que, como ya se mencionó

en la sección 5.4.1, bastará con realizar una simple multiplicación. En otros, sin embargo, se deberán tener en cuenta otras consideraciones. Así pues, hay dos casos a considerar:

- Rutas que cruzan un nodo entrando y saliendo por puertos de la misma dimensión (Propiedad 5.2).
- Rutas que cruzan un nodo entrando y saliendo por puertos de distinta dimensión (Propiedad 5.3).

Propiedad 5.2 *Dado un nodo* $\langle x, y, z \rangle$*, existen* $\frac{(k-1)(k-3)}{4}k^2$ *rutas si* k *es impar, y* $\frac{(k-2)^2}{4}k^2$ *rutas si* k *es par, que cruzan dicho nodo usando el puerto de entrada* P *y el de salida* P'*, donde* $P \neq P'$ *y ambos pertenecen a la misma dimensión, es decir,* $P, P' \in \{X^+, X^-\}$ *o* $P, P' \in \{Y^+, Y^-\}$ *o* $P, P' \in \{Z^+, Z^-\}$*. De otra forma:*

$$R_{X\leftrightarrow X}(\langle x, y, z \rangle) = R_{Y\leftrightarrow Y}(\langle x, y, z \rangle) = R_{Z\leftrightarrow Z}(\langle x, y, z \rangle) = \begin{cases} \frac{(k-1)(k-3)}{4}k^2 & \textit{si k impar} \\ \frac{(k-2)^2}{4}k^2 & \textit{si k par} \end{cases}$$

Demostración: Se demuestra para la dimensión X, siendo el proceso de demostración totalmente similar para las dimensiones Y y Z.

En la figura 5.2 se muestra un subconjunto de nodos del toro $3D$, todos ellos pertenecientes a la dimensión X. Para simplificar, se usa sólo el dígito correspondiente a la dimensión X para identificar a cada nodo.

k impar

Considerando sólo los nodos de la dimensión X, es decir teniendo en cuenta sólo la parte de la ruta que ocupa la dimensión X, se puede observar en la figura 5.2 que existen:

- 0 rutas con origen en el nodo $(x - \frac{k-1}{2}) mod\ k$ que cruzan el nodo x entrando por X^- y saliendo por X^+
- 1 ruta con origen en el nodo $(x - \frac{k-1}{2} + 1) mod\ k$ que cruzan el nodo x entrando por X^- y saliendo por X^+
- 2 rutas con origen en el nodo $(x - \frac{k-1}{2} + 2) mod\ k$ que cruzan el nodo x entrando por X^- y saliendo por X^+
- ...

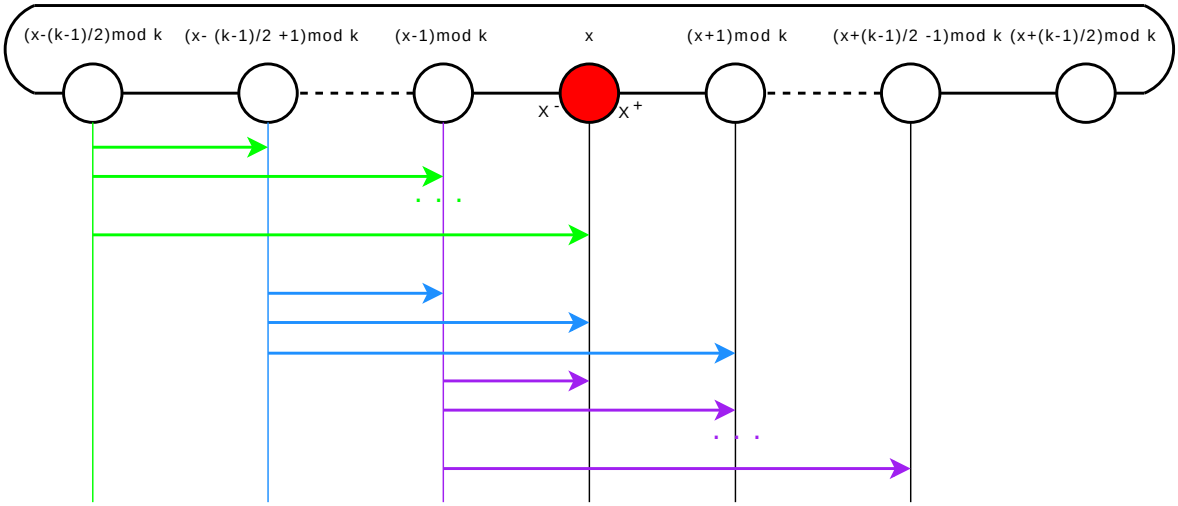


Figura 5.2: Rutas que cruzan por un nodo dado en la dimensión X.

- $\frac{k-1}{2} - 1$ rutas con origen en el nodo $(x-1) mod\ k$ que cruzan el nodo x entrando por X^- y saliendo por X^+

Por tanto, hay un total de $\sum_{i=1}^{\frac{k-1}{2}-1} i$ rutas que atraviesan el nodo x usando X^- como puerto de entrada y X^+ como puerto de salida, suponiendo que estas rutas tienen su origen y destino en la dimensión X. Como k es impar se tiene el mismo número de rutas cuando la entrada al nodo es por el puerto X^+ y la salida por el puerto X^-.

Ahora bien, las rutas que cruzan por un nodo en la dimensión X pueden recorrer también las otras dos dimensiones. Es decir, el nodo destino puede ser cualquiera de los k^2 nodos a los que se puede llegar a través de las dimensiones Y y Z.

Por tanto, el número total de rutas que cruzan el nodo $\langle x, y, z\rangle$ entrando y saliendo por puertos distintos de la dimensión X viene dado por:

$$\begin{aligned} R_{X\leftrightarrow X}(\langle x,y,z\rangle) &= R_{X^-\to X^+}(\langle x,y,z\rangle) + R_{X^+\to X^-}(\langle x,y,z\rangle) = 2\sum_{i=1}^{\frac{k-1}{2}-1} ik^2 = \\ &= 2\frac{(\frac{k-1}{2}-1)(\frac{k-1}{2})}{2}k^2 = \frac{k-3}{2}\frac{k-1}{2}k^2 = \frac{(k-1)(k-3)}{4}k^2 \end{aligned}$$

k par

Mediante un razonamiento similar se puede obtener el número de rutas que cruzan el nodo $\langle x, y, z\rangle$ entrando y saliendo por puertos distintos de la dimensión X cuando k es par. Sólo hay que tener en cuenta que el número de rutas cruzando el nodo $\langle x, y, z\rangle$ entrando por el puerto X^- y saliendo por el puerto X^+ no es el mismo que el número de rutas que lo cruzan entrando por el puerto X^+ y saliendo por el puerto X^-.

Así pues, para k par se obtiene lo siguiente:

$$R_{X\leftrightarrow X}(\langle x, y, z\rangle) = R_{X^-\rightarrow X^+}(\langle x, y, z\rangle) + R_{X^+\rightarrow X^-}(\langle x, y, z\rangle) = \sum_{i=1}^{\frac{k}{2}-1} ik^2 + \sum_{i=1}^{\frac{k}{2}-2} ik^2$$

Por tanto, el número total de rutas que cruzan el nodo $\langle x, y, z\rangle$ entrando y saliendo por puertos distintos de la dimensión X viene dado por:

$$\begin{aligned} R_{X\leftrightarrow X}(\langle x, y, z\rangle) &= \left(\sum_{i=1}^{\frac{k}{2}-1} i + \sum_{i=1}^{\frac{k}{2}-2} i\right) k^2 = \left(2\sum_{i=1}^{\frac{k}{2}-2} i + \left(\frac{k}{2}-1\right)\right) k^2 = \\ &= \left(2\frac{\left(\frac{k}{2}-1\right)\left(\frac{k}{2}-2\right)}{2} + \left(\frac{k}{2}-1\right)\right) k^2 \\ &= \left(\frac{k}{2}-1\right)\left(\frac{k}{2}-2+1\right) k^2 = \left(\frac{k}{2}-1\right)^2 k^2 = \frac{(k-2)^2}{4} k^2 \end{aligned}$$

Para las dimensiones Y y Z el tratamiento es similar. A lo largo de la dimensión Y ocurre lo mismo que lo indicado para la dimensión X. Además una ruta que cruza por un nodo $\langle x, y, z\rangle$ de la dimensión Y puede tener su origen en cualquiera de los k nodos de la dimensión X desde los que se puede llegar a $\langle x, y, z\rangle$. Y desde $\langle x, y, z\rangle$, al saltar a la dimensión Z también se pueden alcanzar k nodos en dicha dimensión.

Y para el caso de la dimensión Z, la ruta puede tener su origen en k^2 nodos de las dimensiones X y Y. En definitiva:

$$R_{X\leftrightarrow X}(\langle x, y, z\rangle) = R_{Y\leftrightarrow Y}(\langle x, y, z\rangle) = R_{Z\leftrightarrow Z}(\langle x, y, z\rangle) = \begin{cases} \frac{(k-1)(k-3)}{4} k^2 & \text{si k impar} \\ \frac{(k-2)^2}{4} k^2 & \text{si k par} \end{cases}$$

□

Propiedad 5.3 *Dado un nodo $\langle x, y, z\rangle$, el número de rutas que cruzan dicho nodo entrando por el puerto P y saliendo por el puerto P', con $P, P' \in \mathcal{P}$ y pertenecientes a diferentes dimensiones, viene dado por*

$$R_{P\to P'}(\langle x, y, z\rangle) = \begin{cases} 0 & \textit{si } P \to P' \textit{ no está permitida} \\ D_s^P(\langle x, y, z\rangle) \times D_d^{P'}(\langle x, y, z\rangle) & \textit{si } P \to P' \textit{ está permitida} \end{cases}$$

Demostración: Si el algoritmo de encaminamiento no permite la transición $P \to P'$, ninguna ruta usará P' inmediatamente después de haber usado P. Por tanto ninguna ruta cruza por el nodo $\langle x, y, z\rangle$ entrando por P y saliendo por P'.

Para las transiciones $P \to P'$ permitidas, el nodo origen de una ruta que cruza el nodo $\langle x, y, z\rangle$ entrando por P puede ser cualquiera de los que pueden alcanzar el nodo $\langle x, y, z\rangle$ llegando a éste por el puerto P, es decir, cualquier nodo perteneciente al conjunto $N_s^P(\langle x, y, z\rangle)$. De igual forma, el nodo destino de esas rutas, que salen del nodo $\langle x, y, z\rangle$ usando P', puede ser cualquiera de los que se pueden alcanzar desde el nodo $\langle x, y, z\rangle$ a través del puerto P', es decir, cualquier nodo perteneciente al conjunto $N_d^{P'}(\langle x, y, z\rangle)$. Por tanto, el número total de rutas que cruzan el nodo $\langle x, y, z\rangle$ usando como puerto de entrada P y de salida P', se obtiene mediante el producto

$$\text{card}(N_s^P(\langle x, y, z\rangle)) \times \text{card}(N_d^{P'}(\langle x, y, z\rangle)) = D_s^P(\langle x, y, z\rangle) \times D_d^{P'}(\langle x, y, z\rangle)$$

□

Aplicando la propiedad 5.3 a todas las situaciones que significan realizar un cambio de dimensión cuando se cruza el nodo $\langle x, y, z\rangle$ se obtiene el número de rutas que cruzan por dicho nodo para cada una de esas situaciones (tabla 5.5).

5.4.6. Evaluación de las configuraciones

Conocido el número de rutas que cruzan un nodo $\langle x, y, z\rangle$ usando cada posible par de puertos, se puede calcular el número de rutas que cruzan el enlace interno que une las dos tarjetas del nodo para cada una de las diez configuraciones posibles. Para una configuración dada, las parejas de puertos que implican el uso del enlace que une las dos tarjetas son aquellas que se forman con un puerto de una tarjeta y otro puerto de la otra, siempre que el uso consecutivo de esos puertos esté permitido por el algoritmo de encaminamiento. Veamos esto con la configuración A.

$$R_{X^-\to Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)k^2}{4} & \text{si k par} \end{cases}$$

$$R_{X^-\to Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{k^3}{4} & \text{si k par} \end{cases}$$

$$R_{X^+\to Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)^2}{4}k & \text{si k par} \end{cases}$$

$$R_{X^+\to Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)k^2}{4} & \text{si k par} \end{cases}$$

$$R_{X^-\to Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{si k impar} \\ \dfrac{(k-2)k}{4} & \text{si k par} \end{cases}$$

$$R_{X^-\to Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{si k impar} \\ \dfrac{k^2}{4} & \text{si k par} \end{cases}$$

$$R_{X^+\to Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{si k impar} \\ \dfrac{(k-2)^2}{4} & \text{si k par} \end{cases}$$

Tabla 5.5: Número de rutas que cruzan el nodo $\langle x,y,z\rangle$.

$$R_{X^+\to Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{si k impar} \\ \dfrac{(k-2)k}{4} & \text{si k par} \end{cases}$$

$$R_{Y^-\to Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)k^2}{4} & \text{si k par} \end{cases}$$

$$R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{k^3}{4} & \text{si k par} \end{cases}$$

$$R_{Y^+\to Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)^2}{4}k & \text{si k par} \end{cases}$$

$$R_{Y^+\to Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{si k impar} \\ \dfrac{(k-2)k^2}{4} & \text{si k par} \end{cases}$$

Tabla 5.5: (Cont.) Número de rutas que cruzan el nodo $\langle x,y,z\rangle$.

La configuración A asigna los puertos de una tarjeta a los enlaces (X^+, Y^+, Z^+) y los de la otra tarjeta a los enlaces (X^-, Y^-, Z^-) (figura 5.1). Una ruta usa las dos tarjetas para cruzar un nodo si la pareja de puertos que utiliza dicha ruta es alguna de las siguientes (donde el primer puerto de la pareja indica el puerto de entrada y el segundo el puerto de salida):

$$(X^+,X^-)\ (X^+,Y^-)\ (X^+,Z^-)\ (Y^+,Y^-)\ (Y^+,Z^-)\ (Z^+,Z^-)$$
$$(X^-,X^+)\ (X^-,Y^+)\ (X^-,Z^+)\ (Y^-,Y^+)\ (Y^-,Z^+)\ (Z^-,Z^+)$$

Las demás combinaciones, permitidas por el algoritmo de encaminamiento, significan entrar al nodo por un puerto que pertenece a una tarjeta y salir por otro puerto que también pertenece a la misma tarjeta, y por tanto no serán consideradas pues no introducen sobrecarga de comunicación a efectos del estudio que se está realizando. Estas parejas de puertos son las siguientes:

$$(X^+, Y^+)\ (X^+, Z^+)\ (Y^+, Z^+)\ (X^-, Y^-)\ (X^-, Z^-)\ (Y^-, Z^-)$$

En este caso, el número de rutas que cruzan un nodo $\langle x, y, z\rangle$ usando el enlace interno que une las dos tarjetas del nodo, usando la configuración A ($\{X^+, Y^+, Z^+\}|\{X^-, Y^-, Z^-\}$) es:

$$\begin{aligned} R_A(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + \\ &+ R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\ &+ R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \\ &= \cdots = \begin{cases} \frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) & \text{si } k \text{ impar} \\ \frac{1}{4}(3k^4 - 8k^3 + 6k^2 + 4k + 4) & \text{si } k \text{ par} \end{cases} \end{aligned}$$

Siguiendo el mismo razonamiento, se puede calcular el número de rutas que usan el enlace interno para el resto de configuraciones. En las tablas 5.6 y 5.7 se incluyen los resultados para k par y k impar, respectivamente.

Configuración	Número de rutas que usan las dos tarjetas al cruzar un nodo
A	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2 + 4k + 4)$
B, F	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2)$
C, I	$\frac{1}{4}(k^4 + 2k^3 - 4k^2 - 2k + 4)$
D	$\frac{1}{4}(k^4 - 4k^2 + 4)$
E	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2 - 4k + 4)$
G	$\frac{1}{4}(k^4 + 4k^2 - 8k + 4)$
H, J	$\frac{1}{4}(k^4 + 2k^3 - 8k^2 + 6k)$

Tabla 5.6: Número de rutas que cruzan un nodo usando las dos tarjetas para cada configuración (k par).

Configuración	Número de rutas que usan las dos tarjetas al cruzar un nodo
A, B, E, F	$\frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2)$
C, H, I, J	$\frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2)$
D, G	$\frac{1}{4}(k^4 - k^2 - 4k + 4)$

Tabla 5.7: Número de rutas que cruzan un nodo usando las dos tarjetas para cada configuración (k impar).

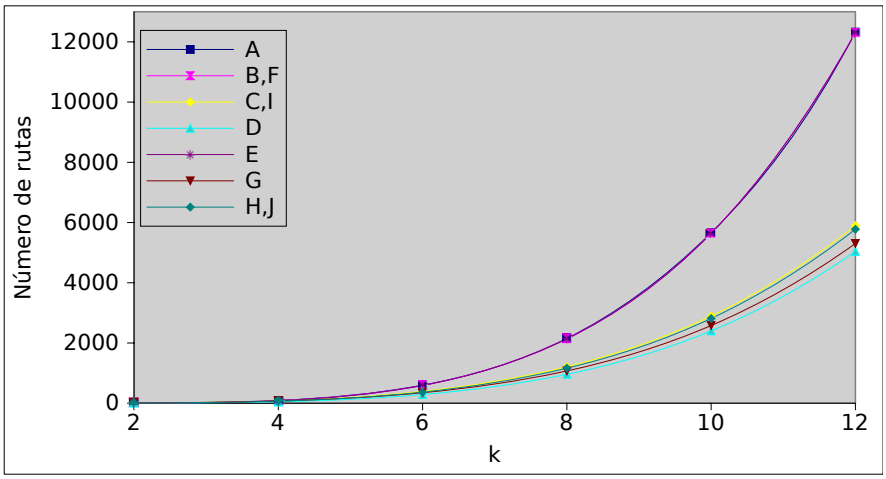
5.4.7. Análisis de los resultados

Operando adecuadamente con las expresiones recogidas en las tablas 5.7 y 5.6, se puede determinar cuál es la configuración óptima en cada caso (k impar y k par), es decir, aquella o aquellas configuraciones que minimizan el número de rutas que usan el enlace interno. Gráficamente, en la figura 5.3 puede observarse la evolución del número de rutas para cada configuración. Así, se tiene que [4]:

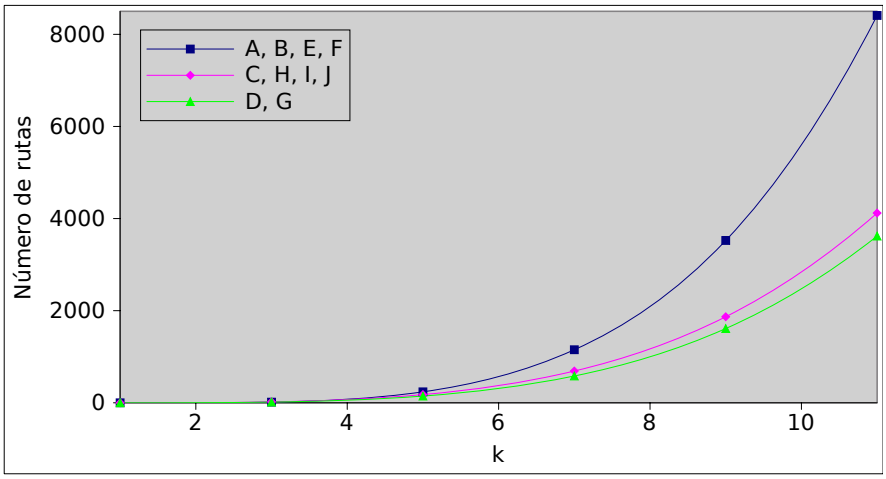
- Para k impar, se tiene que:
  - Si $k = 3$, las configuraciones A, B, F y E son óptimas.
  - Si $k \geq 5$, las configuraciones D y G son óptimas.
- Para k par, se tiene que:
  - Si $k = 2$, las configuraciones D y E son óptimas.
  - Si $k \geq 4$, sólo la configuración D es óptima.

  Cabe destacar que cuando la distancia entre una pareja de nodos origen-destino es la misma tomando un sentido u otro en una dimensión dada, se ha tomado la decisión de establecer la ruta hacia el sentido positivo. Pues bien, si se eligiese el sentido contrario, se podría demostrar con un estudio similar que la configuración óptima en ese caso hubiese sido la G.

La figura 5.4 muestra las dos configuraciones que ofrecen las mejores prestaciones.

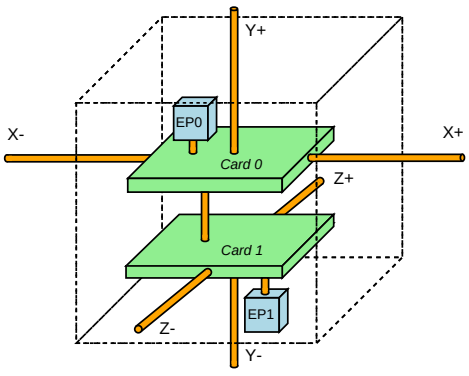


(a) k par

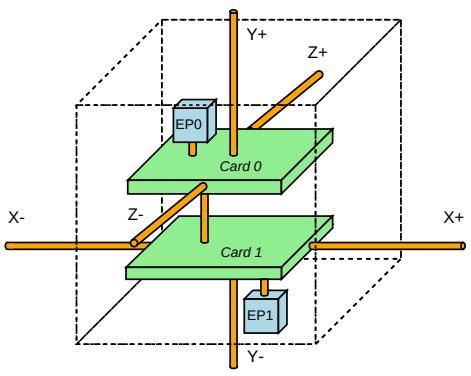


(b) k impar

Figura 5.3: Número de rutas que cruzan un nodo usando las dos tarjetas.



(a) (X^+, Y^+, X^-)–(Y^-, Z^+, Z^-) para k par e impar.



(b) (X^+, Y^-, X^-)–(Y^+, Z^+, Z^-) para k impar.

Figura 5.4: Configuraciones óptimas.

CAPÍTULO 6

ENCAMINAMIENTO EN TOROS 3DT

El algoritmo de encaminamiento es el mecanismo que determina el camino que debe seguir un mensaje en la red para alcanzar su destino a partir del nodo fuente en el que se ha generado. En ciertos casos, pueden darse condiciones que dificultan la tarea del encaminamiento, como *deadlock*, *livelock* o *starvation* (sección 4.2.6). De no ser tratado correctamente, el *deadlock* es un problema inherente a las topologías n-cubos k-arios toroidales. Este problema se agrava en los toros $3DT$ debido a que el enlace interno de los nodos es compartido por todas las dimensiones del toro, no pudiendo ser solucionado mediante las técnicas básicas usadas en el resto de toros.

Por ello, este capítulo se dedica al estudio y eliminación del *deadlock* en las topologías $3DT$. En primer lugar se presenta el algoritmo de encaminamiento DOR adaptado a toros $3DT$ (sección 6.1). A continuación, se presenta un estudio sobre los ciclos que aparecen en nuestra topología (sección 6.2), para finalizar explicando cómo eliminar estos ciclos y con ellos el *deadlock*, en la sección 6.3.

6.1. ALGORITMO DOR ADAPTADO A LA TOPOLOGÍA TORO $3DT$

En el caso de la topología toro $3DT$, cada EP necesita un identificador formado por un dígito por cada dimensión del toro (X, Y y Z, en este orden) además de un cuarto dígito para indicar el EP de proceso dentro del nodo. Se avanza por las dimensiones que sea necesario, y cuando ya se han recorrido las tres dimensiones, se comprueba si el mensaje está destinado al EP actual o al EP vecino, encaminando el mensaje hacia el NIC o hacia el enlace interno, respectivamente. Por último, se ha de comprobar si el puerto de salida seleccionado pertenece al EP, y en caso contrario, el puerto seleccionado será el enlace interno. En las tablas 6.1 y 6.2 se puede observar el algoritmo DOR adaptado en pseudo-código. Señalar que en la función $sentidoAnillo()$, que determina el sentido para llegar al nodo destino dentro de una anillo cualquiera, la letra D se utiliza para referirse a una dimensión cualquiera.

Entrada: nodo actual $\langle x_c, y_c, z_c, ep_c \rangle$,
nodo destino $\langle x_d, y_d, z_d, ep_d \rangle$
Salida: puerto salida p

si $x_d \neq x_c$ **entonces**
  $p = sentidoAnillo(x_c, x_d)$
sino, si $y_d \neq y_c$ **entonces**
  $p = sentidoAnillo(y_c, y_d)$
sino, si $z_d \neq z_c$ **entonces**
  $p = sentidoAnillo(z_c, z_d)$
sino, si $ep_d \neq ep_c$ **entonces**
  $p = link\_interno$
sino
  $p = NIC$
fin si
si $p \in LINKS(ep_c)$ **entonces**
  devolver p
sino
  devolver $link\_interno$
fin si

Tabla 6.1: Algoritmo de encaminamiento DOR para toros $3DT$.

Entrada: dígito actual d_cur,
dígito destino d_des
Salida: puerto salida ($D^+,\ D^-$)

$aux = (d\_des - d\_cur) mod\ k$
si $aux > k/2$ **entonces**
  $aux = aux - k$
fin si
si $aux \geq 0$ **entonces**
  devolver D^+
sino
  devolver D^-
fin si

Tabla 6.2: Función $sentidoAnillo()$.

6.2. ESTUDIO DE CICLOS EN TORO 3DT

Como se comentó en la sección 4.2.6.1, la mayoría de los algoritmos de encaminamiento deterministas basan sus propiedades de libertad de bloqueo en la ausencia de bucles en el grafo de dependencia de canales [9]. Sin embargo, conforme crece el tamaño de la red, más crece el grafo de dependencia de canales, por lo que la tarea de eliminar los ciclos en el grafo se vuelve harto complicada. Por ello, se ha preferido averiguar el motivo por el que aparecen estos ciclos en la topología toro $3DT$ para actuar consecuentemente y eliminar el *deadlock* en la topología.

6.2.1. Tipos de tráfico en el enlace interno

Desafortunadamente, en la topología toro $3DT$ aparecen nuevos ciclos en la red que no se presentan en las topologías toro $3D$ tradicionales. Esto es debido al enlace interno, el cual puede ser usado por un mensaje independientemente de la dimensión por la que se esté desplazando. En algunos casos, el mensaje usará el enlace interno como parte del anillo de una dimensión, en otros casos para un cambio entre dimensiones. En la figura 6.1 se puede

observar los anillos correspondientes a cada dimensión usando la configuración D[1] y como, en este caso, el enlace interno forma parte del anillo de la dimensión Y.

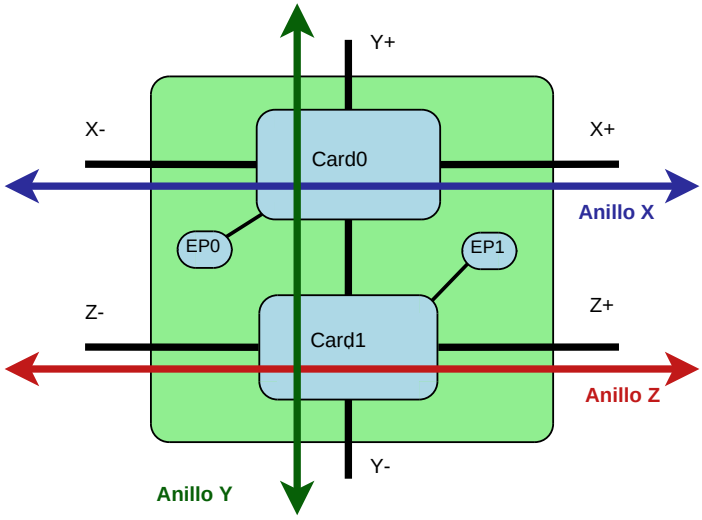


Figura 6.1: Anillos correspondientes a cada dimensión en un nodo cualquiera de la red.

Concretamente, si se analiza el uso del enlace interno, se pueden distinguir 3 casos (figura 6.2), dependiendo del destino del mensaje tras usar el enlace interno.

1.- El mensaje debe usar el enlace interno para ser inyectado en una dimensión d donde el enlace interno no forma parte del anillo correspondiente a la dimensión d. En la figura 6.2 se puede observar como un mensaje proveniente de la dimensión X o Y debe usar el enlace interno para ser inyectado en la dimensión Z (línea punteada roja).

2.- El mensaje usa el enlace interno como parte del anillo de una dimensión d. El mensaje puede estar recorriendo la dimensión d antes de usar el enlace interno, o ser inyectado desde otra dimensión. En la figura 6.2 se puede observar que tanto un mensaje proveniente de la dimensión X, como un mensaje que ya circula por la dimensión Y, deben usar el enlace interno para salir por el puerto Y^- del nodo (línea punteada amarilla).

3.- El destino del mensaje es el EP conectado a la otra tarjeta del nodo. En este caso, el mensaje puede provenir de cualquiera de los puertos de entrada de la tarjeta actual (línea punteada azul).

[1]De aquí en adelante, en el resto de ejemplos será utilizada la configuración D, dado que es óptima para $k \geq 4$, y es a partir de estos valores de k cuando la topología toro $3DT$ obtiene ventaja sobre el toro $2D$.

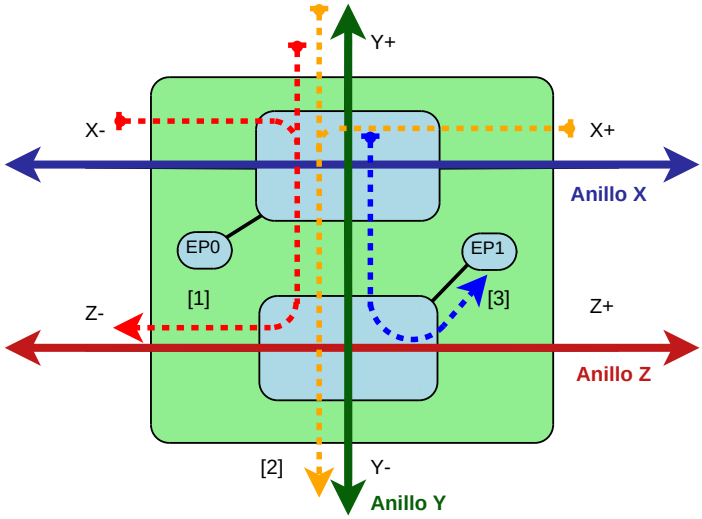


Figura 6.2: Posibles usos del enlace interno.

6.2.2. Tipos de ciclos

A partir del uso del enlace interno que realizan los mensajes en la red, se han podido identificar dos tipos de ciclos en los que participa el enlace interno:

A.- Varios mensajes usan el enlace interno como parte del anillo de una dimensión d. Sin el tratamiento adecuado, puede aparecer bloqueos en cualquier anillo, siendo ésta la razón de la aparición del *deadlock* en los topologías toroidales tradicionales. Este tipo de ciclo es debido al tráfico del tipo 2. En el ejemplo 6.1 se muestra con más detalle una situación en la que surge un bloqueo debido a este motivo.

B.- Varios mensajes utilizan en su recorrido varios enlaces internos para acceder a una nueva dimensión y también para alcanzar el EP destino. Este tipo de ciclo aparece debido al tráfico de los casos 1. y 3. En el ejemplo 6.2 se muestra con más detalle una situación en la que surge un bloqueo debido a este motivo.

Ejemplo 6.1 *Sea una red toro $3DT$, con 2 nodos en la dimensión Y y puertos configurados de acuerdo a la configuración D. Supongamos que:*

- *El $EP0$ del nodo $\langle x, 0, z\rangle$ envía un mensaje al $EP0$ del nodo $\langle x, 1, z\rangle$, y viceversa.*
- *El $EP1$ del nodo $\langle x, 0, z\rangle$ envía un mensaje al $EP1$ del nodo $\langle x, 1, z\rangle$, y viceversa.*

en cuyo caso se producen ciclos en la red, pudiéndose llegar a una situación de bloqueo. En la figura 6.3 puede observarse gráficamente esta situación.

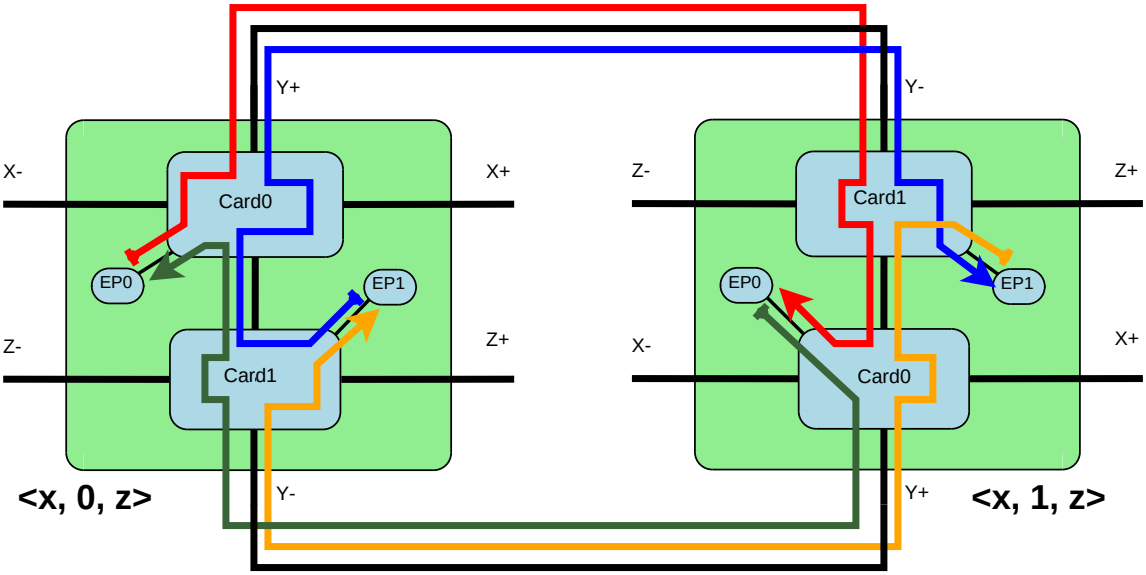


Figura 6.3: Posible situación de bloqueo debido al uso del enlace interno como parte de la dimensión Y.

Ejemplo 6.2 *Sea una red toro $3DT$ de tamaño cualquiera y puertos configurados de acuerdo a la configuración D. Supongamos que:*

- *El $EP1$ del nodo $\langle x, y, z\rangle$ envía un mensaje al $EP0$ del nodo $\langle x+1, y, z+1\rangle$.*
- *El $EP1$ del nodo $\langle x+1, y, z+1\rangle$ envía un mensaje al $EP0$ del nodo $\langle x, y, z\rangle$.*

en cuyo caso se producen ciclos en la red, pudiéndose llegar a una situación de bloqueo. En la figura 6.4 puede observarse gráficamente esta situación.

6.3. ELIMINACIÓN DE CICLOS EN TOROS 3DT

Una vez que se conoce el motivo por el que surgen los ciclos en la red, se procederá a su eliminación. Dos de las técnicas más ampliamente utilizadas en n-cubos k-arios son los canales virtuales [9] y el control de flujo con mecanismo de la burbuja [6]. En los enlaces externos del nodo pueden aplicarse estas técnicas de igual forma que se aplican en un toro tradicional, pero es necesario realizar modificaciones para eliminar los bloqueos en el enlace interno. En las secciones 6.3.1 y 6.3.2 se presentan las modificaciones realizadas al algoritmo de encaminamiento DOR para eliminar el bloqueo, mediante canales virtuales y el algoritmo de la burbuja, respectivamente.

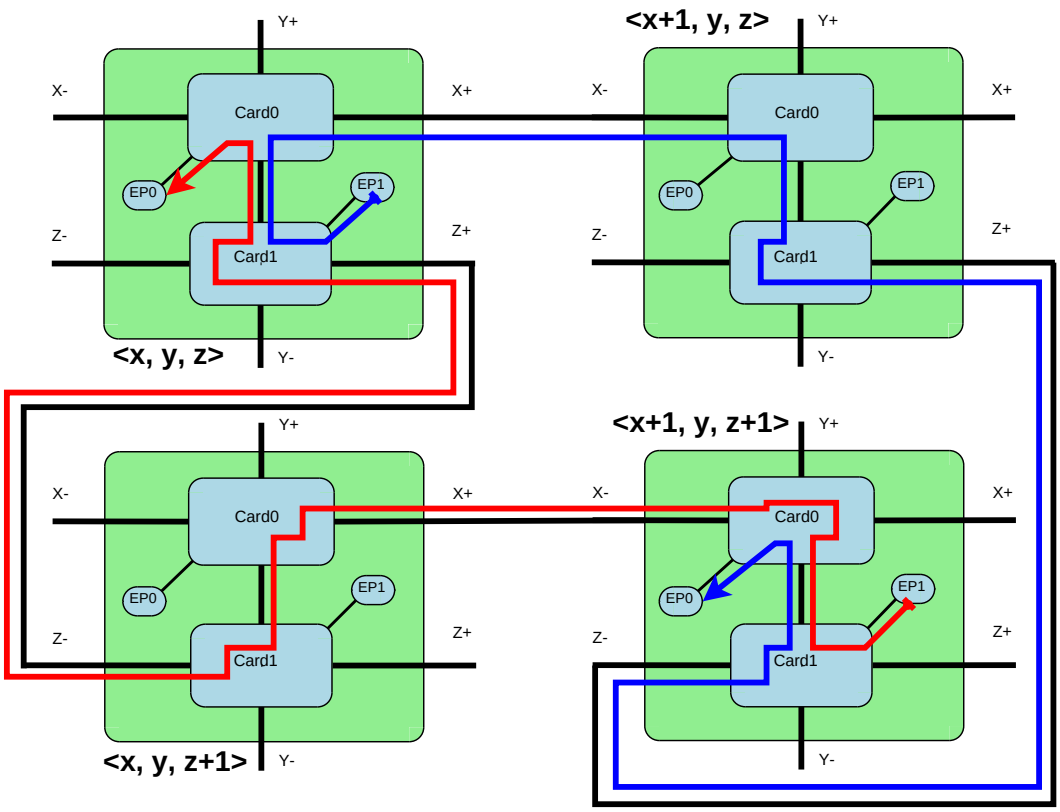


Figura 6.4: Posible situación de bloqueo debido al uso del enlace interno para cambiar de dimensión y para llegar al EP de destino.

6.3.1. Canales virtuales

Como se ha comentado anteriormente, aparecen nuevos ciclos en la red debido a que todas las dimensiones hacen uso del enlace interno. Para romper estos ciclos, es necesario multiplexar el enlace interno usando canales virtuales y tratar el flujo de datos de cada dimensión por separado. También es necesario encaminar los mensajes destinados al EP vecino por un canal virtual exclusivo, para evitar los ciclos de tipo B. De esta forma, se requerirán los siguientes canales virtuales:

- Un canal virtual para los mensajes destinados a una dimensión donde el enlace interno no forma parte del anillo (tipo 1.). En el algoritmo de encaminamiento modificado para la configuración D se ha escogido el canal virtual 0 para este tipo de tráfico.
- Dos canales virtuales para los mensajes destinados a una dimensión donde el enlace interno forma parte del anillo (tipo 2.). Para seleccionar cuál de dos canales será utilizado por el mensaje, se utiliza el mismo criterio que en los enlaces externos del nodo. Es decir, si el destino en la dimensión actual es mayor que el nodo actual se utiliza el primer canal virtual, en caso contrario se utiliza el segundo canal virtual destinado a esta dimensión. Los canales virtuales 1 y 2 han sido los escogidos para este caso en el algoritmo de encaminamiento modificado para la configuración D.

Entrada: nodo actual $\langle x_c, y_c, z_c, ep_c \rangle$,
nodo destino $\langle x_d, y_d, z_d, ep_d \rangle$
Salida: puerto salida p, canal virtual vc

si $x_d \neq x_c$ **entonces**
  $p = sentidoAnillo(x_c, x_d)$
sino, si $y_d \neq y_c$ **entonces**
  $p = sentidoAnillo(y_c, y_d)$
sino, si $z_d \neq z_c$ **entonces**
  $p = sentidoAnillo(z_c, z_d)$
sino, si $ep_d \neq ep_c$ **entonces**
  $p = link\_interno$
  $vc = 3$ // *tipo 3.*
sino
  $p = NIC$
fin si
si $p \notin LINKS(ep_c)$ **entonces**
  $p = link\_interno$
  si $p \neq Y^+$ y $p \neq Y^-$ **entonces**
    $vc = 0$ // *tipo 1.*
  sino, si $vc = Up\_Links$ **entonces**
    $vc = 1$ // *tipo 2.*
  sino
    $vc = 2$ // *tipo 2.*
  fin si
fin si

Entrada: dígito actual d_cur, dígito destino d_des
Salida: puerto salida p, canal virtual vc

$aux = (d\_des - d\_cur) mod\ k$
si $aux > k/2$ **entonces**
  $aux = aux - k$
fin si
si $aux \geq 0$ **entonces**
  $p = D^+$
sino
  $p = D^-$
fin si
si $d\_des > d\_cur$ **entonces**
  $vc = Up\_Links$
sino
  $vc = Low\_Links$
fin si

Tabla 6.3: Modificaciones del algoritmo de encaminamiento (izquierda) y la función *SentidoAnillo* (derecha) para que sea libre de bloqueo usando canales virtuales y la configuración D.

- Un canal virtual para los mensajes destinados al EP vecino (tipo 3.). Para este caso, se ha escogido el canal virtual 3 en el algoritmo modificado para la configuración D.

De esta forma, el grafo de dependencia de canales es libre de ciclos, por lo cual no podrán aparecer bloqueos en la red. Es importante señalar que el número de canales virtuales necesarios dependerá de la configuración del nodo utilizada. En los casos en que los puertos de las tres dimensiones se encuentren en distintas tarjetas, serán necesarios $3 \times 2 + 1 = 7$ canales virtuales (configuraciones A, B, E y F), mientras que en el resto sólo son necesarios $1 + 1 \times 2 + 1 = 4$ canales (configuraciones C, D, G, H, I y J).

En la tabla 6.3 pueden observarse las modificaciones realizadas en el algoritmo de encaminamiento, mientras en la figura 6.5 (a) se muestra gráficamente el uso de los canales virtuales dependiendo de cada caso, siempre considerando la configuración D.

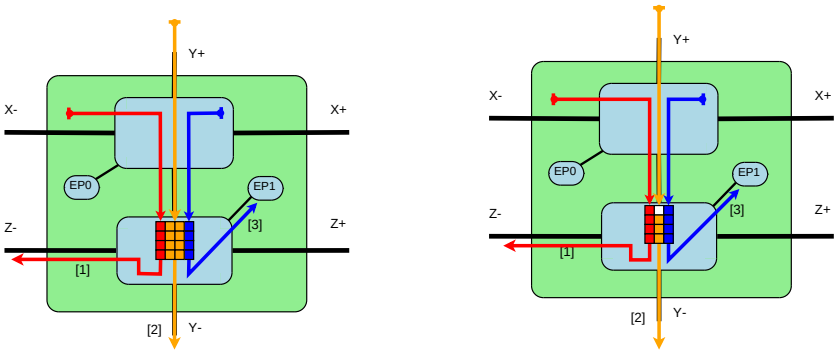
6.3.2. Control de flujo con mecanismo de la burbuja

En este caso, también se hace necesario multiplexar el canal para tratar por separado los flujos de datos de cada dimensión, aunque son necesarios menos canales virtuales para asegurar la ausencia de bloqueos en la red. Así, son necesarios:

- Un canal virtual para las dimensiones donde el enlace interno no forma parte del anillo (tipo $1.$). En este caso, el acceso al enlace interno no se considera cambio de dimensión, por lo que es necesario aplicar la burbuja. El cambio de dimensión se realiza al acceder al puerto externo del nodo desde el enlace interno. En el algoritmo de encaminamiento modificado para la configuración D se ha escogido el canal virtual 0 para este tipo de tráfico.
- Un canal virtual por dimensión si el enlace interno forma parte del anillo (tipo $2.$). El enlace interno es considerado un enlace más de la dimensión, por lo que es necesario aplicar la burbuja si accedemos desde otra dimensión diferente. En el algoritmo de encaminamiento modificado se ha utilizado el canal virtual 1 para este tipo de tráfico.
- Un canal virtual para los mensajes destinados al EP vecino (tipo $3.$). No se aplica la burbuja, ya que el posterior destino del mensaje es el NIC de la tarjeta vecina y dejar el hueco sólo serviría para reducir las prestaciones. Para este caso, se ha escogido el canal virtual 2 en el algoritmo modificado para la configuración D.

Este mecanismo nos asegura la ausencia de bloqueos en la red. En este caso, si los puertos de las tres dimensiones se encuentren en distintas tarjetas, son necesario $3{\times}1+1=4$ canales virtuales (configuraciones A, B, E y F), mientras que en el resto de configuraciones son necesarios $1+1+1=3$ canales (configuraciones C, D, G, H, I y J).

En la tabla 6.4 pueden observarse las modificaciones realizadas en el algoritmo de encaminamiento, mientras en la figura 6.5 (b) se muestra gráficamente el uso de los canales dependiendo del tipo de tráfico, en ambos casos para la configuración D.



(a) Canales Virtuales (b) Burbuja

Figura 6.5: Soluciones para eliminar el *deadlock* usando la configuración D.

Entrada: nodo actual $\langle x_c, y_c, z_c, ep_c \rangle$, nodo destino $\langle x_d, y_d, z_d, ep_d \rangle$
Salida: puerto salida p, canal virtual vc

si $x_d \neq x_c$ **entonces**
  $p = sentidoAnillo(x_c, x_d)$
sino, si $y_d \neq y_c$ **entonces**
  $p = sentidoAnillo(y_c, y_d)$
sino, si $z_d \neq z_c$ **entonces**
  $p = sentidoAnillo(z_c, z_d)$
sino, si $ep_d \neq ep_c$ **entonces**
  $p = link\_interno$
  $vc = 2$ // *tipo 3.*
  $bub =$ **falso**
sino
  $p = NIC$
  $bub =$ **falso**
fin si
si $p \notin LINKS(ep_c)$ **entonces**
  $p = link\_interno$
  si $p \neq Y^+$ y $p \neq Y^-$ **entonces**
    $vc = 0$ // *tipo 1.*
    $bub =$ **falso**
  sino
    $vc = 1$ // *tipo 2.*
  fin si
fin si

Entrada: dígito actual d_cur, dígito destino d_des, puerto entrada p_{in}
Salida: puerto salida p_{out}, uso burbuja bub

$aux = (d\_des - d\_cur) mod\ k$
si $aux > k/2$ **entonces**
  $aux = aux - k$
fin si
si $aux \geq 0$ **entonces**
  $p_{out} = D^+$
sino
  $p_{out} = D^-$
fin si
si $p_{in} = D^+$ o $p_{in} = D^-$ **entonces**
  $bub =$ **falso**
sino, si ($p_{out} = Y^+$ o $p_{out} = Y^-$) y $p_{in} = link\_interno$ **entonces**
  $bub =$ **falso**
sino
  $bub =$ **verdadero**
fin si

Tabla 6.4: Modificaciones del algoritmo de encaminamiento (izquierda) y la función *SentidoAnillo* (derecha) para que sea libre de bloqueo usando el mecanismo de la burbuja y la configuración D.

CAPÍTULO 7

EVALUACIÓN DE LA TOPOLOGÍA

En este capítulo se presenta una evaluación de prestaciones de la topología toro $3DT$. En primer lugar, se han comparado las distintas configuraciones posibles del nodo en la topología para corroborar el estudio teórico y comparar la diferencia de rendimiento entre las distintas configuraciones. Tras esto, se realizará un estudio de las mismas características comparando la topología toro $3DT$ con la topología toro $2D$.

La evaluación de las topologías se ha realizado mediante simulación. En la sección 7.1 se describen las características del simulador usado, mientras que en la sección 7.2 se incluye el conjunto de métricas utilizado para la evaluación de prestaciones. Por último, se muestran los experimentos realizados y los resultados obtenidos para la evaluación de las distintas configuraciones del toro $3DT$ y la comparación de las topologías $3DT$ y $2D$ en las secciones 7.3 y 7.4, respectivamente.

7.1. MODELO DEL SISTEMA

Como se ha comentado anteriormente, la evaluación de la topología toro $3DT$ se ha realizado mediante un simulador. Para modelar las diversas topologías a evaluar, se ha implementado un conmutador de 5 puertos (4 puertos para interconectar los conmutadores y un puerto para conectar el conmutador con su EP correspondiente), que comparte la mayor parte de sus características en todas las topologías. Dependiendo de la topología y solución escogida para eliminar el *deadlock*, únicamente variará de un conmutador a otro la función de encaminamiento, la organización de los buffers de los canales físicos y si el encaminamiento implementa el mecanismo de la burbuja o se basa en el uso de canales virtuales.

Concretamente, se ha implementado un conmutador tipo IQ (*Input Queued*) [19, 22]. En este tipo de conmutador sólo hay buffers en los puertos de entrada del mismo (figura 7.1), siendo utilizados estos buffers para retener los paquetes que en un momento dado no se pueden encaminar al puerto de salida. En nuestro caso, dado que los puertos son bidireccionales, los flits se almacenarán en los buffers al entrar al conmutador.

En cuanto a la técnica de conmutación, se ha escogido *Virtual cut–through*, ya que es una técnica utilizada comúnmente en el ámbito de los supercomputadores. Dado que los EPs no están integrados con el propio conmutador, el área del conmutador destinada a almacenamiento no es un elemento tan restrictivo como en otro tipo de sistemas. Además,

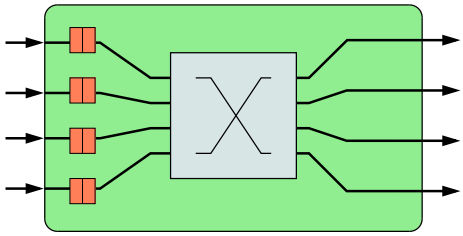


Figura 7.1: Esquema de conmutador IQ.

esta técnica de conmutación no es tan sensible como otras al diámetro de la red, lo cual es ideal en redes con un número tan grande de nodos.

Por otra parte, se ha modelado un control de flujo basado en créditos, mientras que el crossbar interno utiliza un árbitro Round-Robin de dos etapas: en primer lugar, se escoge un puerto de entrada con paquetes que puedan atravesar el crossbar, para después elegir entre uno de los canales virtuales correspondientes a ese puerto (en los casos que sea necesario).

En cuanto al encaminamiento, se ha utilizado el algoritmo DOR habitual para las topologías toro $2D$, mientras que para las topologías toro $3DT$ se han utilizado los algoritmos descritos en la sección 6. Dependiendo de la solución utilizada para eliminar el *deadlock*, el conmutador implementará el mecanismo de la burbuja, o existirán canales virtuales, si son necesarios.

En todos los casos, el tamaño del canal físico es el mismo. En los experimentos realizados, se ha escogido un tamaño de 128 flits y un tamaño de paquete de 4 flits. El número y tamaño de los canales virtuales varía dependiendo de la topología y la solución al *deadlock* implementada:

- **Control de flujo con burbuja**:
  - *Topología toro 2D*:
    - 1 puerto hacia el EP con un único canal virtual de 128 flits.
    - 4 puertos de interconexión con un único canal virtual de 128 flits.
  - *Topología toro 3DT*:
    - 1 puerto hacia el EP con un único canal virtual de 128 flits.
    - 3 puertos de interconexión con un único canal virtual de 128 flits.
    - 1 puerto de interconexión (enlace interno) con 4 canales virtuales[1] de 32 flits.

[1]Aunque sólo son necesarios 3 canales virtuales, se han utilizado 4 canales virtuales para facilitar la implementación. Para el tráfico destinado a una dimensión cuyo anillo no forma parte del enlace interno, se han utilizado dos canales virtuales en lugar de uno sólo, eligiendo el canal virtual de destino en función del puerto de salida de la dimensión.

- **Canales Virtuales**:
  - *Topología toro 2D*:
    - 1 puerto hacia el EP con 4 canales virtuales de 32 flits.
    - 4 puertos de interconexión con 4 canales virtuales de 32 flits.
  - *Topología toro 3DT (conf C, D, G, H, I y J)*:
    - 1 puerto hacia el EP con 4 canales virtuales de 32 flits.
    - 4 puertos de interconexión con 4 canales virtuales[2] de 32 flits.
    - 1 puerto de interconexión (enlace interno) con 4 canales virtuales de 32 flits.
  - *Topología toro 3DT (conf A, B, E y F)*:
    - 1 puerto hacia el EP con 4 canales virtuales de 32 flits.
    - 3 puertos de interconexión con 4 canales virtuales de 32 flits.
    - 1 puerto de interconexión (enlace interno) con 8 canales virtuales[3] de 16 flits.

Por último, se ha modelado una carga de tráfico uniforme, es decir, todos los EPs en la red tienen la misma probabilidad de ser el EP destino de cualquiera de los mensajes generados desde cualquier EP origen. Dicho de otro modo, en una red con N EPs, un mensaje generado en un EP es dirigido a uno de los $N-1$ restantes con igual probabilidad.

7.2. MÉTRICAS PARA LA EVALUACIÓN DE PRESTACIONES

Para realizar la evaluación de prestaciones, en este trabajo se han considerado las siguientes métricas:

- **Productividad media**. Medida en paquetes/ciclo, indica la productividad de la red, es decir, la cantidad de información que la red es capaz de entregar por unidad de tiempo.
- **Latencia media en la red**. Este valor representa la media de los retrasos producidos por la transmisión de los paquetes en la red, medida en ciclos. La latencia de un mensaje mide el número de ciclos que trascurren desde que el mensaje es inyectado en un conmutador de la red desde el NIC asociado al EP origen hasta que es recibido por el NIC asociado al EP destino.
- **Latencia media extremo a extremo**. La latencia extremo a extremo mide el numero de ciclos que trascurren desde que un mensaje es generado por el NIC asociado al EP de origen hasta que es recibido en el NIC asociado al EP destino. Este valor representa la latencia extremo a extremo media en la red.

[2] De nuevo, aunque sólo son necesarios dos canales virtuales, se han escogido cuatro para facilitar la implementación.

[3] En este caso se han utilizado 8 canales virtuales en lugar de 7 para facilitar la implementación, usando el canal virtual extra introducido para el tráfico destinado al NIC del EP vecino.

7.3. EVALUACIÓN DE LAS DISTINTAS CONFIGURACIONES DEL TORO 3DT

En esta sección se va a estudiar la influencia en las prestaciones de la red de la configuración escogida para el nodo de la topología toro $3DT$. A continuación, se presentan los experimentos realizados y los resultados obtenidos, para finalizar la sección con un breve análisis de dichos resultados.

7.3.1. Experimentos realizados

Para la evaluación de las prestaciones de las distintas configuraciones, se ha realizado una batería de pruebas, variando en cada caso el tamaño de la red, la configuración del nodo y la solución empleada para eliminar el *deadlock*. Cada prueba consta de 30 experimentos, siendo los resultados presentados en el siguiente apartado los valores medios de los experimentos de cada uno. Las pruebas se han creado teniendo en cuenta las siguientes consideraciones:

- Se han estudiado las siguientes topologías:
  - Toro $3DT$ $4{\times}4{\times}2$, 64 EPs[4].
  - Toro $3DT$ $4{\times}4{\times}4$, 128 EPs.
  - Toro $3DT$ $5{\times}5{\times}5$, 250 EPs.
- Tratamiento del deadlock:
  - Control de flujo con mecanismo de la burbuja.
  - Canales virtuales.
- Se han realizado pruebas para las 10 configuraciones posibles del nodo (A-J).

7.3.2. Resultados obtenidos

En esta sección se presentan gráficamente los resultados obtenidos. En la figura 7.2 se muestra, para un topología toro $3DT$ $4 \times 4 \times 2$, como evolucionan la productividad, la latencia de red y la latencia extremo a extremo conforme aumenta la tasa de inyección de los EPs de la red, para cada configuración posible del nodo. Los resultados se presentan de forma idéntica en las figuras 7.3 y 7.4 para las pruebas realizadas con 128 y 250 EPs, respectivamente.

[4]Recordar que en cada nodo de la red hay 2 EPs.

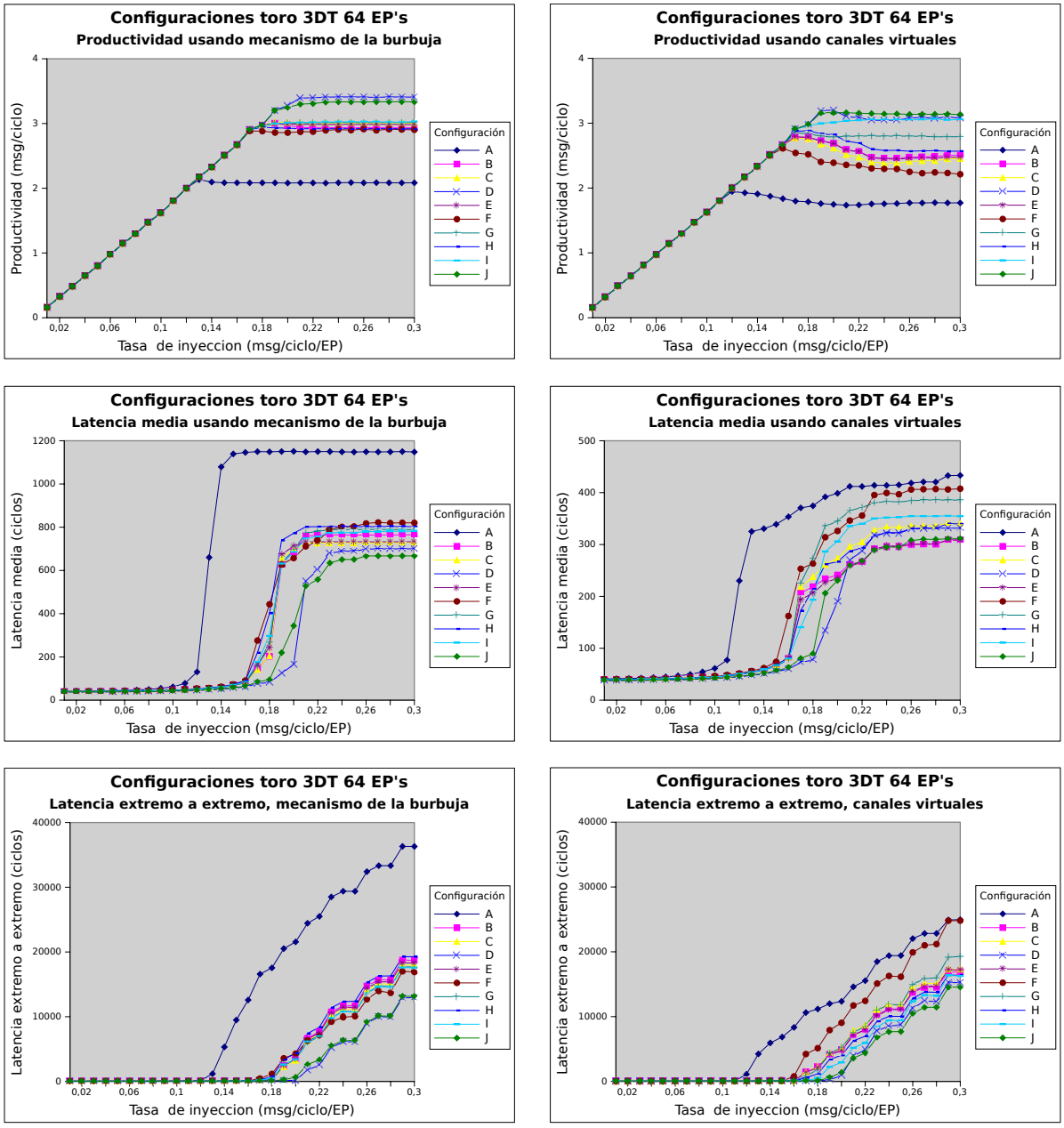


Figura 7.2: Prestaciones obtenidas para las distintas configuraciones de un toro $3DT$ 4×4×2 (64 EPs).

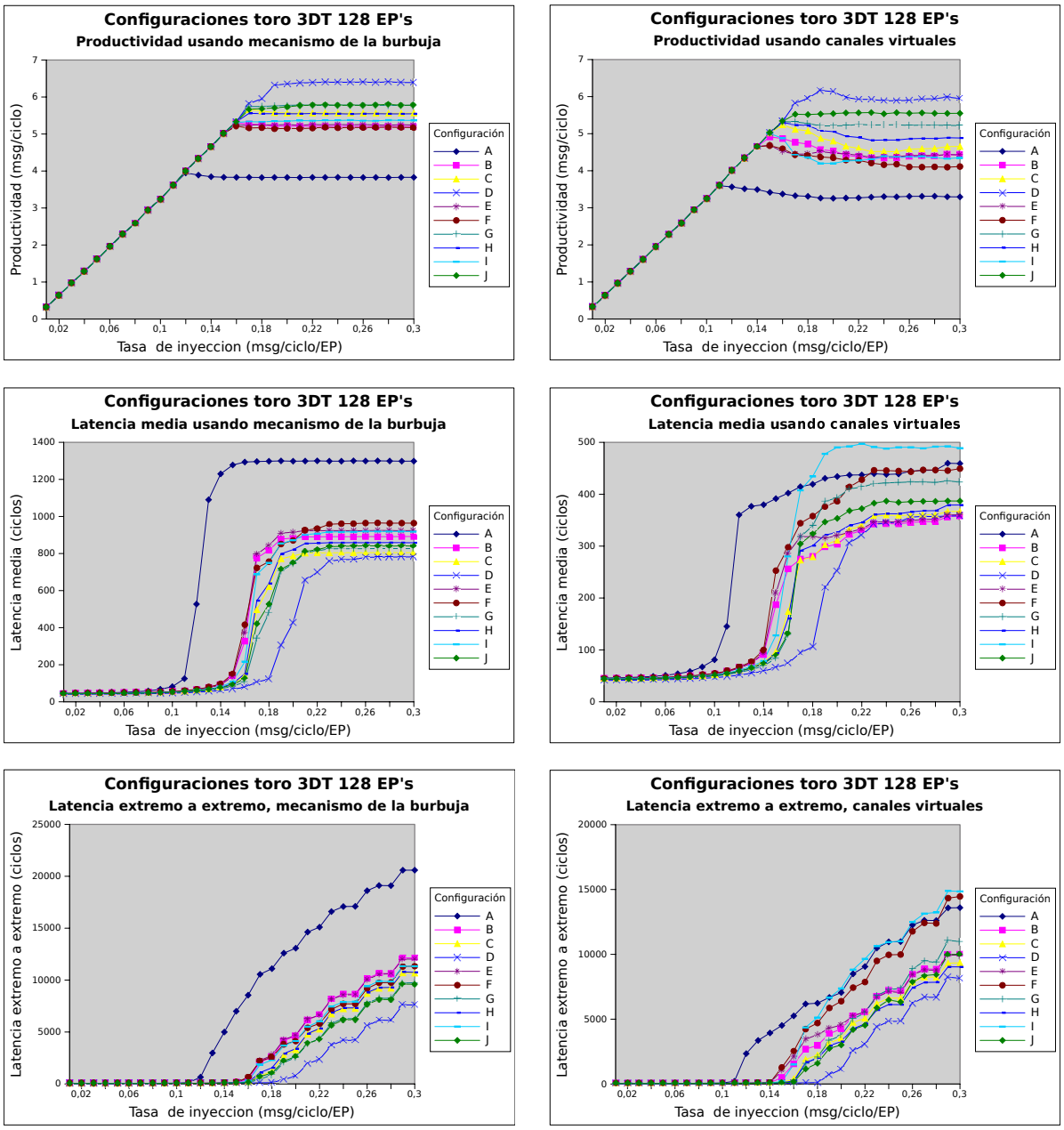


Figura 7.3: Prestaciones obtenidas para las distintas configuraciones de un toro $3DT$ 4×4×4 (128 EPs).

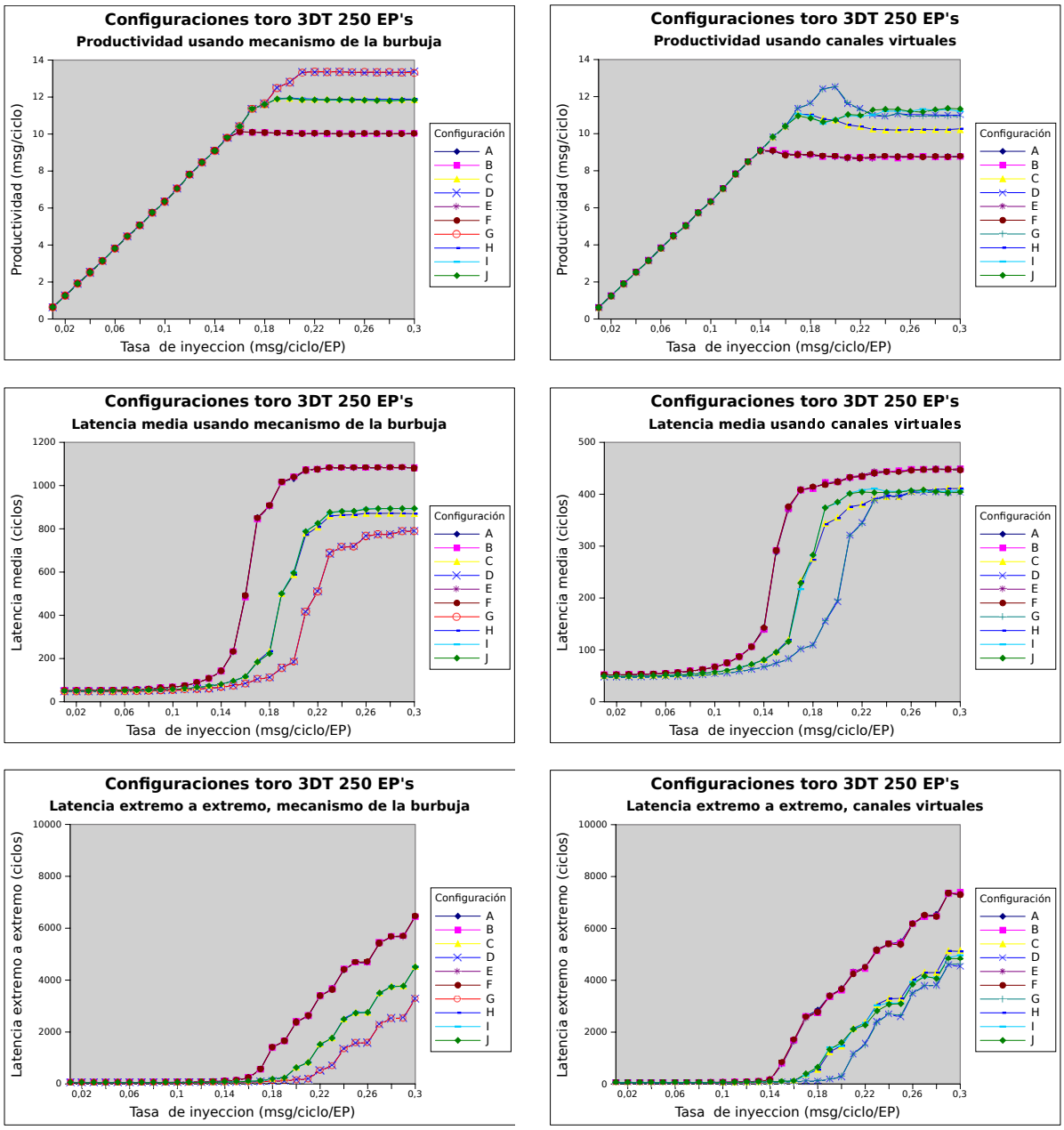


Figura 7.4: Prestaciones obtenidas para las distintas configuraciones de un toro $3DT$ $5{\times}5{\times}5$ (250 EPs).

7.3.3. Análisis de los resultados

En vista de los resultados, se puede observar que generalmente la configuración D es la que obtiene mejores prestaciones, excepto en el toro de $64\ EPs$, donde existen otras configuraciones que se comportan de manera similar. Sin embargo, conforme aumenta el tamaño de la red las diferencias entre configuraciones se hacen más significativas, siendo la D la que ofrece mejores resultados. Estos resultados están en total consonancia con el resultado del estudio teórico presentado en el capítulo 5.

Por ejemplo, mientras que para $128\ EP's$ y usando el mecanismo de la burbuja, la productividad de la configuración D es un $10\,\% \sim 23\,\%$ (dependiendo de la configuración) mayor que en el resto de configuraciones, para $250\ EP's$ la productividad es $15\,\% \sim 33\,\%$ veces mayor. Lo mismo pasa con la latencia de la red y la latencia extremo a extremo, que pasan de reducirse un $5\,\% \sim 20\,\%$ y un $20\,\% \sim 40\,\%$ a reducirse un $12\,\% \sim 28\,\%$ y un $30\,\% \sim 50\,\%$, respectivamente.

Cabe destacar que en el toro $3DT$ de $250\ EPs$ ($k = 5$), la configuración G obtiene prestaciones similares a la configuración D, tal como se esperaba tras realizar el estudio teórico, pues ambas configuraciones son óptimas si k es impar. También se puede observar como el resto de configuraciones se agrupan en dos conjuntos que se comportan de forma similar, tal y como se esperaba, pues tienen en común el número de rutas que cruzan el enlace interno de sus nodos.

Se puede observar como en las topologías que usan canales virtuales, se produce una caída en la productividad de la red cuando ésta llega al punto de saturación, siendo más acusada esta caída conforme aumenta el tamaño de la red. Este problema aparece debido a la congestión en la red y es típico de los conmutadores con este tipo de arquitecturas [5, 18]. Debido a esto, en el toro $3DT$ de $250\ EPs$ que usa canales virtuales, la degradación de prestaciones hace que las configuraciones D y G igualen su productividad al de las configuraciones C, H, I y J, aunque su latencia media y extremo a extremo sigue siendo un $10\,\%$ menor. Este efecto no ocurre en el conmutador que utiliza el mecanismo de la burbuja, manteniendo las prestaciones de las configuraciones D y G por encima del resto.

Teniendo en cuenta lo aquí expuesto, que es lo mismo que se obtuvo con el estudio teórico, en el resto de pruebas que se realicen para la topología toro $3DT$ se utilizará la configuración D.

7.4. COMPARATIVA DEL TORO 3DT FRENTE AL TORO 2D

En esta sección se va a comparar la prestaciones obtenidas por las topologías toro $2D$ y toro $3DT$ para redes con el mismo tamaño de EPs. A continuación, se presenta los experimentos realizados y los resultados obtenidos, para finalizar la sección con un breve análisis de los resultados.

7.4.1. Experimentos realizados

Al igual que en la evaluación de las diversas configuraciones del toro $3DT$, se ha realizado una serie de pruebas formadas por 30 experimentos cada una. En este caso, la batería de pruebas se ha creado teniendo en cuenta las siguientes consideraciones:

- Se han comparado las siguientes topologías:
  - Redes con 64 EPs:
    - Topología toro $2D$ 8×8.
    - Topología toro $3DT$ $4\times4\times2$.
  - Redes con 128 EPs:
    - Topología toro $2D$ 16×8.
    - Topología toro $3DT$ $4\times4\times4$.
  - Redes con 256 (250) EPs:
    - Topología toro $2D$ 16×16.
    - Topología toro $3DT$ $8\times4\times4$.
    - Topología toro $3DT$ $5\times5\times5$ (250 EPs)[5].
  - Redes con 1024 EPs:
    - Topología toro $2D$ 32×32.
    - Topología toro $3DT$ $8\times8\times8$.
- En todos los toros $3DT$ se ha utilizado la configuración D.
- Tratamiento del deadlock:
  - Control de flujo con mecanismo de la burbuja.
  - Canales virtuales.

7.4.2. Resultados obtenidos

En esta sección se presentan gráficamente los resultados obtenidos. En la figura 7.5 se muestran los resultados obtenidos para las topologías $2D$ y $3DT$ con 64 EPs (izquierda) y 128 EPs (derecha), mientras que en la figura 7.6 se muestran las gráficas correspondientes a las topologías con 256 EPs (izquierda) y 1024 EPs (derecha).

[5]Se ha elegido esta topología porque permite construir un toro $3DT$ con el mismo número de nodos por dimensión y con el número de EPs más cercanos a 256 EPs.

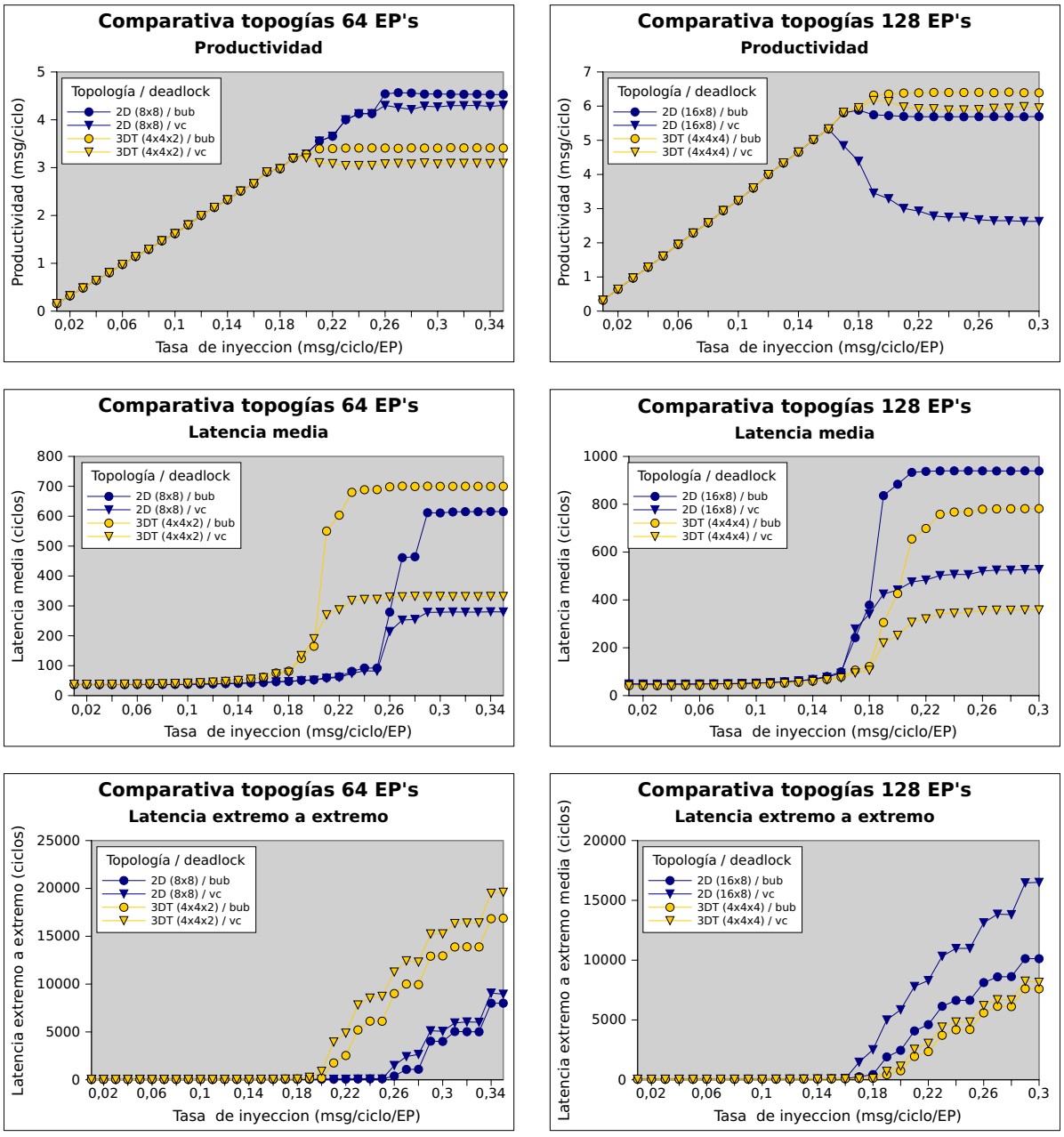


Figura 7.5: Prestaciones obtenidas para las topologías toro $2D$ y $3DT$ con 64 y 128 EPs.

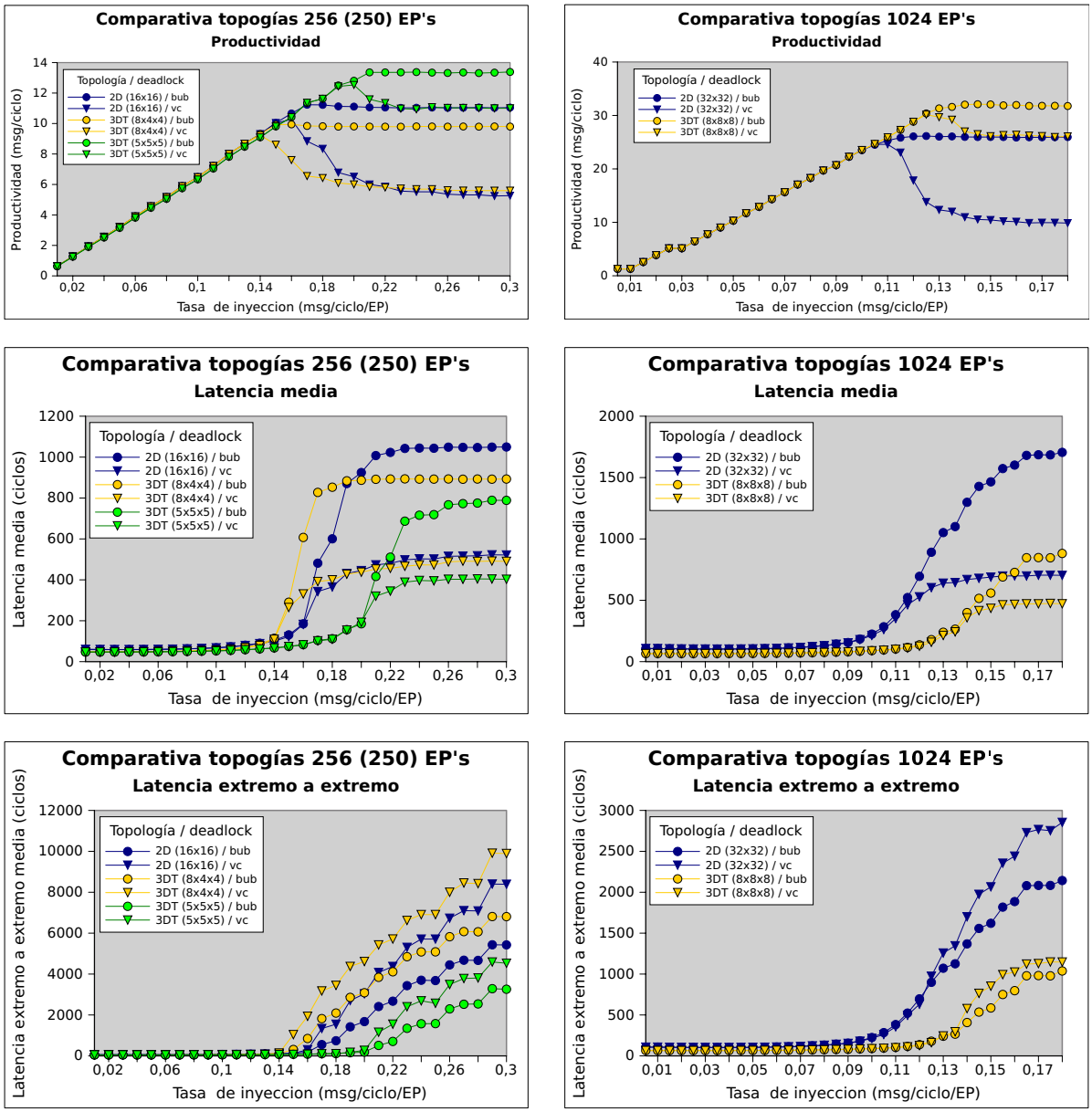


Figura 7.6: Prestaciones obtenidas para las topologías toro $2D$ y $3DT$ con 256 (250) y 1024 EPs.

7.4.3. Análisis de los resultados

Tal y como se esperaba tras realizar el estudio teórico, el toro $3DT$ ofrece mejores prestaciones que el toro $2D$ a partir de redes con más de 128 elementos de proceso ($k \geq 4$). Se puede observar como en la red con 64 EPs la topología toro $2D$ ofrece mejor rendimiento, pero a partir de 128 EPs el toro $3DT$ ofrece mejores resultados, siendo la diferencia entre topologías mayor conforme crece el tamaño de la red. Aunque el toro $3DT$ $8{\times}4{\times}4$ ofrece peores prestaciones que el toro $2D$, debido a que el tamaño de sus dimensiones es diferente, el toro $3DT$ $5 \times 5 \times 5$, aunque con un número ligeramente menor de EPs, ofrece mayor productividad y menor latencia, dado que sus tres dimensiones tienen el mismo tamaño.

En cuanto a la caída de prestaciones experimentada en las topologías que usan canales virtuales para eliminar el *deadlock*, se puede observar como su efecto es mucho mayor en las topologías toro $2D$. A pesar de usar el mismo conmutador en ambos casos, las topologías $3DT$ resisten mucho mejor al efecto negativo de la congestión, siendo la disminución de la productividad mucho menor. Mientras las caída de la productividad en el toro $2D$ es de un $50\,\% \sim 60\,\%$, en el toro $3DT$ sólo experimenta una caída del $10\,\% \sim 20\,\%$.

A su vez, esto hace que haya grandes diferencias entre la productividad obtenida en el toro $2D$ y el toro $3DT$, ya que el toro $2D$ sólo alcanza el $50\,\%$ (256 EPs) y el $40\,\%$ (1024 EPs) de la productividad conseguida por su equivalente $3DT$. Estas diferencias son mucho menores usando el mecanismo de la burbuja, donde el toro $3DT$ obtiene un $15\,\%$ más de productividad que su equivalente $2D$.

Por último, en cuanto al tipo de solución utilizada para eliminar el *deadlock*, el uso de canales virtuales disminuye la latencia media de la red en un $50\,\%$, sin embargo el uso del control de flujo con la burbuja aumenta la productividad en un $20\,\%$ y disminuye la latencia extremo a extremo, aunque la diferencia en esta última métrica se hacen menos significativa conforme aumenta el tamaño de la red. Como se comentó anteriormente, la congestión hace que caigan las prestaciones de la red con canales virtuales, haciendo que estas redes obtengan peor productividad y latencia extremo a extremo que una red con la misma topología y que utilice el control de flujo con la burbuja.

CAPÍTULO 8

CONCLUSIONES Y TRABAJO FUTURO

En este último capítulo se recogen las conclusiones más importantes obtenidas tras la realización del trabajo. Además, se plantean una serie de tareas que se realizarán a continuación del trabajo aquí presentado y que servirán para la posterior realización de la tesis doctoral.

8.1. CONCLUSIONES

Tras analizar los resultados obtenidos en la sección 7, las conclusiones más importantes del estudio se pueden resumir en los siguientes puntos:

- Tal y como se esperaba tras la realización del estudio teórico, **la configuración D (y la G si k es impar) del nodo para el toro** $3DT$ **es la que ofrece mejores prestaciones** en redes con más de 128 EPs, si bien debido a la degradación de prestaciones experimentada en las redes que usan el mecanismo de canales virtuales, muchas otras configuraciones se comportan de manera similar a la configuración D. En cualquier caso, si los conmutadores utilizan el mecanismo de la burbuja, la configuración D será la que obtenga mayor rendimiento.

- Coincidiendo con los resultados obtenidos en el estudio teórico, **la topología toro** $3DT$ **obtiene mejores prestaciones que una topología toro** $2D$, con redes del mismo tamaño o similar, con más de 128 EPs y usando el mismo tipo de conmutadores. Además, las diferencias entre las prestaciones de cada topología se hacen más grandes conforme aumenta el tamaño de la red.

- Usando el mecanismo de la burbuja se obtiene mayor productividad y menor latencia extremo a extremo que usando el mecanismo de canales virtuales, aunque de esta forma se consigue una menor latencia media en la red.

- Si se usa el mecanismo de canales virtuales para eliminar el *deadlock*, **la topología toro** $3DT$ **es más resistente a la degradación de prestaciones causada por la congestión** que su equivalente $2D$ con el mismo número de elementos de proceso, siendo la caída de la productividad mucho menos acusada en las topologías toro $3DT$.

8.2. TRABAJO FUTURO

A continuación, se presentan algunas de las líneas de trabajo que darán continuidad al estudio aquí presentado:

- Para modelar la carga se ha utilizado únicamente un patrón de tráfico uniforme, por lo que resulta de gran interés estudiar el rendimiento del toro $3DT$ y sus configuraciones bajo otros patrones de tráfico, y sobre todo, bajo cargas de tráfico real, generadas a partir de aplicaciones científicas que realicen sus cálculos en un entorno $3D$.
- Estudiar la influencia sobre otros aspectos de la red, como el área o el consumo de las tarjetas, que tiene el uso de las diferentes propuestas que se han expuesto en este trabajo para la construcción de la topología toro $3DT$.
- Desarrollo de algoritmos de encaminamiento adaptativos que tengan en cuenta la estructura interna del nodo, e intenten minimizar el número de rutas que deben usar el enlace que une las dos tarjetas internas.
- Extender el estudio, teórico y practico aquí presentado, para toro n-dimensionales. Con tarjetas de 6 puertos o más, y con el mismo planteamiento expuesto en este trabajo, se pueden construir topologías con mayor número de dimensiones.

Bibliografía

[1] Cluster resources, empresa desarrolladora de TORQUE. http://www.clusterresources.com/.

[2] Sitio web usixml. http://www.usixml.org.

[3] N. Adiga, G. Almasi, Y. Aridor, R. Barik, D. Beece, R. Bellofatto, G. Bhanot, R. Bickford, M. Blumrich, and A. A. Bright. An overview of the blue gene/l supercomputer. In *Supercomputing 2002 Technical Papers*, 2002.

[4] F. J. Andújar, J. A. Villar, , F. J. Alfaro, J. L. Sánchez, and J. Duato. Building 3d torus using low–profile expansion cards. Technical Report DIAB-11-02-3, Department of Computing Systems. University of Castilla-La Mancha, 2011. http://www.dsi.uclm.es/descargas/thecnicalreports/DIAB-11-02-3/diab-11-02-3.pdf.

[5] K. Bolding. Non-uniformities introduced by virtual channel deadlock prevention. Technical report, 1992.

[6] C. Carrion, R. Beivide, J. Gregorio, and F. Vallejo. A flow control mechanism to avoid message deadlock in k-ary n-cube networks. In *High-Performance Computing, 1997. Proceedings. Fourth International Conference on*, pages 322–329, dec 1997.

[7] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. In *in Proc. IEEE INFOCOM*, pages 1048–1056, 1995.

[8] W. Dally and C. Seitz. The torus routing chip. *Journal of Distributed Computing*, 1(3):187–196, Oct. 1986.

[9] W. Dally and C. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Computers, IEEE Transactions on*, C-36(5):547 –553, may 1987.

[10] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[11] D. Dias and J. Jump. Packet Switching Interconnection Networks for Modular Systems. *Computer*, 14(12):43–53, 1981.

[12] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection networks. An engineering approach*. Morgan Kaufmann Publishers Inc., 2003.

[13] R. A. Fisher. *Statistical Methods for Research Workers*. Originally published in Edinburgh by Oliver and Boyd., 1925.

[14] M. Gerla and L. Kleinrock. Flow Control: A Comparative Survey. *IEEE Trans. Com*, (28), 1980.

[15] IBM Blue Gene Team. Overview of the ibm blue gene/p project. *IBM Journal of Research and Development*, 52(1/2), 2008.

[16] C. Inc. Scalable computing at work: Cray xt4 datasheet. `http://www.cray.com/downloads/Cray_XT4_Datasheet.pdf`, 2006.

[17] C. Inc. Cray xt specifications. `http://www.cray.com/Products/XT/Specifications.aspx`, 2009.

[18] C. Izu. Throughput fairness in k-ary n-cube networks. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 137–145, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[19] M. Karol and M. Hluchyj. Queuing in high-performance packet-switching. *IEEE Journal on Selected Areas*, 1:1587–1597, 1998.

[20] P. Kermani and L. Kleinrock. Virtual cut-through: A new computer communication switching technique. *Computer Networks*, 3:267–286, 1979.

[21] C. E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Trans. Comput.*, 34(10):892–901, 1985.

[22] N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick, and M. Horowitz. The tiny tera: A packet switch core. *IEEE Micro*, 17:27–33, 1997.

[23] I. E. Miranda, F. F. Palacín, M. A. L. Sánchez, M. M. Márquez, A. M. R. Chía, A. S. Navas, and C. V. Franco. *Inferencia Estadística: Teoría y problemas*. Servicio de Publicaciones de la Universidad de Cadiz, 2007.

[24] V. Shurbanov, D. Avresky, P. Mehra, and W. Watson. Flow Control in Servernet®Clusters. *J. Supercomput.*, 22(2):161–173, 2002.

[25] Student. The probable error of a mean. *BiometriKa*, VI:1–25, 1908.

[26] T. supercomputer sites. Lista de los 500 supercomputadores más potentes del mundo. `http://www.top500.org`, 2010.

[27] J. A. Villar, F. J. Andújar, F. J. Alfaro, J. L. Sánchez, and J. Duato. An alternative for building high-radix switches: Application for special traffic patterns. Technical Report DIAB-11-02-2, Department of Computing Systems. University of Castilla-La Mancha, 2011. `www.dsi.uclm.es/descargas/thecnicalreports/DIAB-11-02-2/diab-11-02-2.pdf`.

[28] J. A. Villar, F. J. Andújar, F. J. Alfaro, J. L. Sánchez, and J. Duato. An alternative for building high-radix switches: Formalization and configuration methodology. Technical Report DIAB-11-02-1, Department of Computing Systems. University of Castilla-La Mancha, 2011.