

Scalable Switching Fabrics for Internet Routers

William J. Dally

Computer Systems Laboratory, Stanford University
and Avici Systems, Inc.

1. Executive Summary

The exponential growth of the Internet is driving a demand for routers that operate at increasing bit-rates (OC48 to OC192 to OC768) and that have a very large number of ports (10s to 100s to 1000s). To meet this growing demand, routers are needed that can scale with the demand along both the bit-rate and port-count axes. To scale a router to handle a large number of high-speed ports requires a *switching fabric* that is itself economically scalable and that provides the quality of service demanded by latency-sensitive traffic.

A switching fabric that transports packets from input ports to output ports is at the core of any router. Historically, routers have used switching fabrics based on backplane buses and crossbar switches. Buses, however, are not scalable to high bit rates, and crossbars, because their cost grows as the square of the number of nodes, cannot be economically scaled to large numbers of nodes.

A direct interconnection network, like the 3-D torus network used in the Avici TSR, provides a high-performance switching fabric that is economically scalable with a cost that increases linearly with the number of nodes. Torus fabrics can be incrementally expanded or upgraded one node at a time. These networks have high path diversity that enables them to route arbitrary traffic patterns without performance degradation. Moreover, they can be realized with uniformly short fabric channels, reducing cost and enabling the use of modern high-speed signaling technology.

Direct interconnection networks have been used for more than a decade in high-performance supercomputers manufactured by Cray Research and others. This high-end computing experience has proven the scalability, economy, and robustness of the technology. Parallel computer networks, by themselves, however, do not provide the

quality-of-service guarantees needed in an Internet router. They are subject to congestion due to *tree-saturation*, and do not bound the delay of guaranteed-bit-rate packets.

The Avici TSR uses a direct interconnection network with separate *virtual networks* for each output port and for each service class. This design leverages the performance, scalability, and availability proven in the high-end computer world and augments it with the quality-of-service required in a network router. In effect, the virtual networks make the 3-D torus appear as if it were a large, output-queued crossbar switch. Like a crossbar, packets destined to different outputs do not interfere with one another by contending for buffer resources. Also, each guaranteed-bit-rate packet sees a low, bounded delay ($<33\mu\text{s}$) because it never competes with best-efforts packets for resources. The two classes of traffic are isolated on separate virtual networks.

The remainder of this white paper discusses the area of scalable switching fabrics in more detail. The next section gives a brief overview of the Avici TSR switching fabric and describes how it provides economic scalability, robustness, and quality-of-service guarantees. Section 3 describes the properties of the torus network and discusses how it provides scalability, economy, and load balance. The technology of virtual fabric networks is described in Section 4 and the application of these networks to providing a non-blocking fabric with quality-of-service guarantees is described.

2. The Avici TSR Switching Fabric

A switching fabric is characterized by three properties: its physical connection (topology), how packets are forwarded over this topology (routing), and how resources are allocated to packets (flow control). The Avici TSR switching fabric uses a three-dimensional torus topology with each line card carrying one node of the torus. The fabric employs source routing with randomization to balance load across alternate paths. The buffers and channels in the fabric are allocated using virtual-channel flow control with a separate virtual channel for each network output port.

This set of design choices offers six significant advantages for a network switch/router:

1. **Economical Scalability:** The 3-D torus topology can be scaled to hundreds of nodes (560 nodes for a 1st generation TSR) with a cost that is linear with the number of nodes and with global bandwidth that increases as nodes are added. In contrast, the

bandwidth of a bus-based fabric does not increase as nodes are added, and the cost of a crossbar fabric increases as the square of the number of nodes, making them uneconomical for more than a handful of nodes. Unlike many other multi-hop networks, the torus exploits physical locality: nodes are connected to their physical neighbors. This facilitates economical interconnects using mostly backplane wiring and keeps signal paths short enabling reliable high-speed signaling.

2. **Incremental Extensibility:** The TSR network can be expanded one line card at a time¹. Because all active components of the fabric are carried on the line cards, the customer need only buy as much fabric as needed. A customer can also incrementally upgrade a switch to a higher speed fabric (with 2nd generation TSR modules) one line card at a time. In contrast, an entire crossbar must be purchased or upgraded as a unit, even if only a small fraction of the ports are populated, and indirect networks (such as butterflies) must be expanded in powers of the network radix², k .
3. **Load balance:** The 3-D torus fabric of the TSR offers a high-degree of *path diversity*. That is, there are many different paths from node A to node B. For example, in a 512-node 8×8×8-torus the average packet can choose from among 90 different 6-hop paths from its source to its destination. The TSRs source routing algorithm distributes packets across these routes to balance the load on the network channels even when the traffic pattern is highly unbalanced. In contrast, most indirect networks (such as butterflies) have no path diversity, offering only a single path from A to B. In these networks, a non-uniform traffic pattern can overload a small set of network channels causing the network to slow to a fraction of its capacity.
4. **Fault tolerance:** Path diversity also gives the TSR network fault tolerance. The network automatically reconfigures around channel or node failures by restricting the set of paths to avoid the failed components. . With a minimum of two edge-disjoint

¹ As a fabric is expanded, not all configurations are regular 3-D tori. However, the source routing protocol is able to handle the irregular networks as well. Also, to provide two edge-disjoint paths between all pairs of line cards, the fabric must be expanded two line cards at a time.

² The *radix* of a butterfly network is the number of input ports and output ports on each fabric switch. Most butterflies are built from 2×2 switches and hence have a radix of 2.

paths between every pair of nodes in the network, the TSR is able to reconfigure around all single component failures and many multiple component failures. All components are hot-pluggable so the fabric can be repaired, expanded, or upgraded without interrupting service. Networks without path diversity are unable to reconfigure around the failed channels or nodes.

5. **Non-Blocking³**: Because the TSR provides a separate *virtual channel* for every output port, packets destined for output A never share a buffer with packets destined for output B. Thus, a packet destined for A cannot block a packet destined for B by holding a buffer it requires. Packets destined for A and B do share physical channel bandwidth. However, fair bandwidth arbitration and headroom on bandwidth provisioning ensure that packets directed to each output receive the bandwidth they require.
6. **Low, Bounded Delay for Constant-Bit-Rate Traffic**: The TSR provides a separate set of virtual channels for guaranteed-bit-rate traffic. Because this traffic never competes with best-efforts traffic for buffers or channel bandwidth, the TSR is able to deliver CBR traffic with a low-bounded delay of $33\mu\text{s}$.

The remainder of this white paper describes in more detail how the TSR design provides economic, scalable performance with high availability and the quality-of-service required for critical traffic.

3. The 3-D torus topology offers incremental scalability, high path diversity, and uniformly short wires

3.1. Basic properties of the torus network

In a three-dimensional torus topology, as illustrated in Figure 1, each node is assigned a three-coordinate address (x, y, z) and is connected by channels in both directions to six neighbors with addresses $(x \pm 1, y \pm 1, z \pm 1)$. The addition and subtraction are modular so that the nodes on one edge of the network are connected to the nodes on the opposite edge. The network may have a different *radix* (number of nodes in a dimension), k_d , in

each dimension, d . The $4 \times 3 \times 2$ network of Figure 1 has $k_x = 4$, $k_y = 3$, and $k_z = 2$. The generation 1 Avici TSR can be scaled to a maximum configuration of $14 \times 8 \times 5$ (560 nodes). Each channel in a generation 1 TSR has a bandwidth of 10Gb/s, four times the OC48 input line rate.

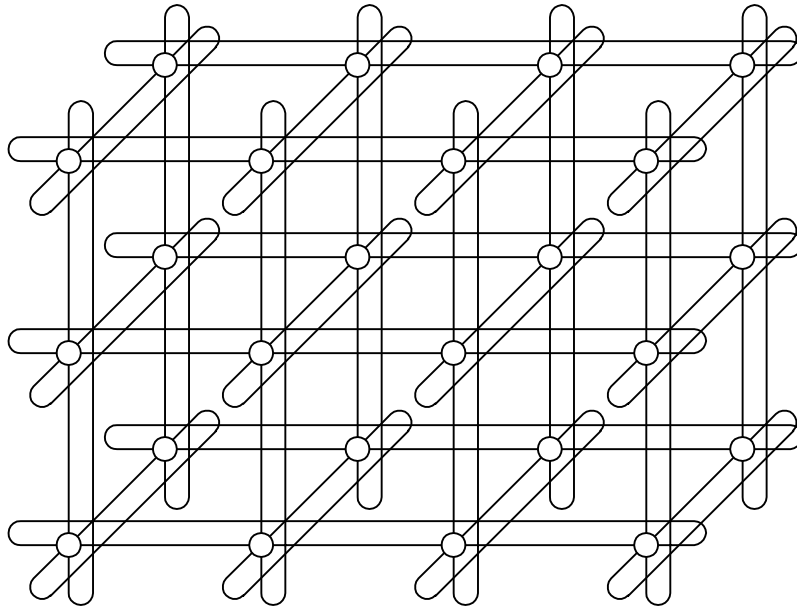


Figure 1: A 4x3x2 torus topology

The torus topology has been widely studied [Dally90a, Dally90b, Agarwal91, DYN97]. Because of its scalability, low latency, high throughput, and high path diversity, it has been the network of choice for manufacturers of scalable parallel computers including Cray Research [KessSchw93, ScottThor94, ScottThor96], and Intel [Carbonaro96]. While packet routers present a somewhat different set of constraints and challenges to the fabric designer, the use of torus networks in packet routers leverages the years of research, design, and operating experience with torus fabrics in scalable parallel computers.

Two key parameters of any topology are its average distance, D , and its worst-case channel load, γ_{max} . Average distance is a major component in determining the latency of a network. In a torus, on average each packet travels one quarter the way around the

³ A *non-blocking* circuit switch is one that can route any permutation from input to output. This definition is not relevant to a packet switch like the TSR. A *non-blocking* packet switch is one in which packets destined to output A cannot block packets destined to a different output, B, by holding buffers or channels required by the B packets.

cycle in each dimension. Thus we have $D = \sum_{i=1}^n \frac{k_i}{4}$, or, for a regular torus, $D = \frac{nk}{4}$.

For example in a 512-node $8 \times 8 \times 8$ torus, the average packet takes two hops in each dimension giving $D = 6$.

The worst-case channel load is the maximum ratio of channel bandwidth to input bandwidth over all of the channels in the network, $\gamma_{\max} = \max_c \left(\frac{\lambda_c}{\lambda_{in}} \right)$. Channel load is a major factor in determining the throughput or bandwidth of a network. For a torus, the channel load in each dimension, i , is $\gamma_i = \frac{k_i}{8}$, so $\gamma_{\max} = \max_i \left(\frac{k_i}{8} \right)$. For example, in a 512-node $8 \times 8 \times 8$ torus, the channels are uniformly loaded with $\gamma_{\max} = \gamma = 1$. That is, for a uniform traffic pattern, applying a load of λ_{in} bits/s to each input places a load of $\lambda_c = \lambda_{in}$ bits/s on each channel. As we shall see below, because of the path diversity of the network, this network also achieves a uniformly low channel loading for non-uniform traffic.

An aggregate measure of network bandwidth is the *bisection bandwidth* of a network, the bandwidth available across a minimal cut that evenly divides the network. That is, if you cut the network in half by passing a plane through its midpoint. The bisection bandwidth is the bandwidth across this plane. For a torus network, the bisection bandwidth is $B = \frac{4b_c N}{k_{\max}}$ for an N -node network with a maximum radix of k_{\max} . For a regular torus network, this reduces to $B = 4b_c N^{\frac{2}{3}}$. Thus, as one adds nodes to a torus network, the aggregate bandwidth increases as the $2/3$ power of the number of nodes⁴. For example, a single-rack $2 \times 4 \times 5$ TSR network has a bisection bandwidth of 320Gbits/s (32 10Gbit/s channels). Expanding this to a maximum-size $14 \times 8 \times 5$ system increases the bisection bandwidth to 1.6Tbits/s (160 10Gbit/s channels).

⁴ One can achieve linear scaling of bandwidth with nodes by using an *express-cube* topology [Dally91]. However, for the range of network sizes required by the TSR, scaling bandwidth as the $2/3$ power of the number of nodes is adequate.

Another bandwidth measure is *speedup*, $S = \frac{b_c}{\gamma_{\max} \lambda_{in}}$, which is the ratio of actual channel bandwidth, b_c , to required channel bandwidth, $\gamma_{\max} \lambda_{in}$. Speedup is the bandwidth *headroom* that is required to handle overhead, fragmentation, contention, and load imbalance. Because of these demands on bandwidth, a fabric will not operate with a speedup of 1. For a network with high path diversity, and hence little load imbalance, a speedup of 2 is usually adequate. With the TSR's bandwidth ratio of $\frac{b_c}{\lambda_{in}} = 4$, an $8 \times 8 \times 8$ torus has a speedup of 4, and a maximally configured TSR (with $k_x = 14$) has a speedup of 2.3. For uniform traffic, speedup can be calculated from bisection bandwidth as

$S = \frac{2B}{N\lambda_{in}}$ where the 2 in the numerator accounts for the fact that half of the traffic in the denominator crosses the network bisection. For a torus network, this gives

$$S = \left(\frac{8}{k_{\max}} \right) \left(\frac{b_c}{\lambda_{in}} \right).$$

As an example of these performance measures, consider an $8 \times 8 \times 8$ node torus network with OC192 input channels $\lambda_{in} = 10^{10}$ bits/s and $b_c = 4 \times 10^{10}$ bits/s network channels. The network has an I/O bandwidth of 5.1 Tbits/s (5.1 Tbits/s of input and 5.1 Tbits/s of output). A plane through the middle of the network cuts 256 channels giving a bisection bandwidth of $B = 10.2$ Tbits/s. Because only half of the input bandwidth needs to cross the network bisection, the speedup is $S = 4$.

3.2. The torus network can be packaged with short wires giving high bandwidth and low cost

A torus network can be packaged with uniformly short wires by folding the torus as shown in Figure 2 for a two-dimensional 4×4 torus. Compared to Figure 1, the nodes of each cycle are *folded* so that the nodes from one side of the cycle are interleaved with the nodes from the other side of the cycle. With folding, each node connects to either an immediately adjacent node (at the edges) or to a node next to an adjacent node. Thus all channels can be realized on uniformly short wires.

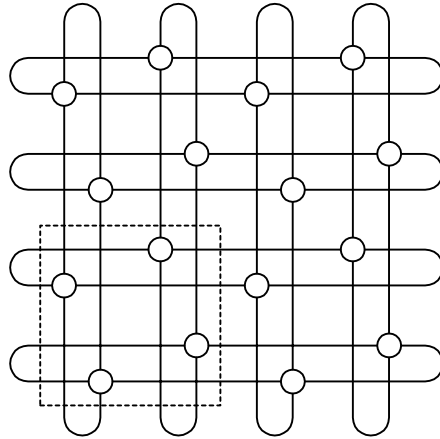


Figure 2: Folded Torus Networks

A topology that can be realized using uniformly short wires gives significant advantages in cost and performance. Short wires can be realized in backplanes, reducing cost by eliminating the need for costly cables and connectors. Also, short wires can be operated at very high bit rates. For a given conductor geometry and signaling system, bandwidth is proportional to the inverse square of the wire length, $B = kL^{-2}$ [DLAPT98]. If a channel is wire-limited operating at a bandwidth B , making the channel twice as long reduces its bandwidth to $B/4$.

Networks that require long wires, like butterflies, Benes networks, and binary n -cubes⁵ have difficulty offering scalable, economical performance. They must either operate their channels at low bit rates or use expensive optical signaling technology to overcome the distance limitation of electrical signaling. Compared to the direct backplane to backplane connections of a folded torus, a cabled connection costs about four times as much per pin, and an optical connection costs over ten times as much. In addition to the cost of components, networks with optical channels also dissipate an order of magnitude more power per signal⁶ and have complex cabling requirements.

Figure 3 shows how the torus network used in the Avici TSR is packaged using uniformly short wires. Figure 3(a) shows the packaging of a single backplane (20 nodes) and Figure 3(b) shows the packaging of a $6 \times 4 \times 5$, 120-node, system. Each TSR backplane (Figure 3(a)) holds 20 line cards organized into four *quadrants* of five line

⁵ A binary n -cube is sometimes called a *hypercube*.

⁶ A good Gb/s electrical driver dissipates 25mW per signal. In contrast, a typical Gb/s fiber optic link dissipates more than 250mW per signal.

cards each. The backplanes are entirely passive. The fabric routers, the active components of the switching fabric, are contained on the line cards. The line cards in each quadrant are connected via backplane conductors in a 5-cycle in the z-dimension. The x- and y-dimension channels from the line cards are brought to the edge of the backplane and connect to the corresponding line card on an adjacent backplane.

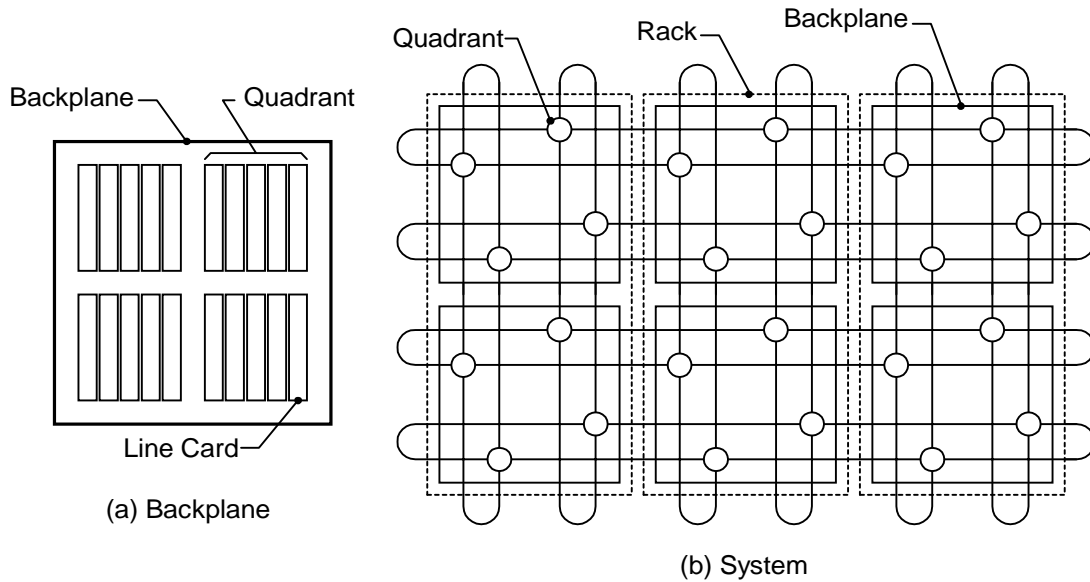


Figure 3: Packaging of the TSR Torus Network

Figure 3(b) shows how connections between adjacent backplanes realize the folded torus topology. Each circle in the figure represents a quadrant (five line cards), and each line represents ten channels (five channels in each direction). As backplanes are assembled in a system, two per rack, the x and y channels of one backplane are connected to the corresponding channels of the adjacent backplanes using a flex-PC jumper. A set of loopback connectors along the edges of the machine completes the torus connection⁷.

With this arrangement, the TSR can be scaled to a size of $14 \times 4 \times 5$, 280 line cards, without using cables, only jumpers between backplanes. All channels in such a system are less than 1m in length. In a fully-expanded $14 \times 8 \times 5$, 560 line card, system, one set of short cables is used to carry the y-dimension channels between two rows of racks.

⁷ For ease of expansion, the loopback connectors are omitted from the top and right edge of a TSR. A special connection in the z-dimension between the quadrants on a backplane prevents this omission from affecting performance.

3.3. The torus network has scalable bandwidth and can be expanded and upgraded one line card at a time

As line cards are added to the torus network of the TSR, the bisection bandwidth of the network scales to carry the traffic from the additional cards. This is in contrast to constant-bandwidth switching fabrics such as buses that cannot be scaled beyond a fixed (usually small) number of nodes because their fixed bandwidth cannot handle the additional traffic.

Figure 4 shows how the capacity⁸ of the torus network in the TSR scales as line cards (nodes) are added to the network. For reference, the figure also shows the I/O bandwidth of the network (2.5 Gb/s per port)⁹. The capacity of the TSR scales from 80Gb/s with 2 nodes to 3.2Tb/s with 320 or more nodes. The capacity follows the $N^{2/3}$ scaling law with three discontinuities due to packaging constraints. Bandwidth is flat at 320Gb/s from 8 to 20 nodes because the 2×2 x-y bisection remains constant as the z-dimension is populated from $2 \times 2 \times 2$ to $2 \times 2 \times 5$. From 100 nodes to 160 nodes bandwidth is flat at 1.6Tb/s because the y-z bisection remains constant at 4×5 as the network is populated from $5 \times 4 \times 5$ to $8 \times 4 \times 5$. Finally network bisection bandwidth remains constant at 3.2Tb/s above 320 nodes because the y-z bisection remains constant at 8×5 as the network is populated from $8 \times 8 \times 5$ to $14 \times 8 \times 5$. Even with these discontinuities, the speedup of the network, $S \geq 4$ up to 320 nodes and remains above 2 beyond 560 nodes.

⁸ The capacity is the amount of input traffic that the network can carry before a fabric link becomes loaded to 100% of capacity. It is twice the bisection bandwidth of the network.

⁹ I/O bandwidth is the input bandwidth or output bandwidth of the fabric, i.e., the line rate times the number of ports. This is how much traffic the fabric can carry. Some manufacturers overcount by reporting the sum of the input and output bandwidth, effectively counting each bit twice, once on the way in and once on the way out.

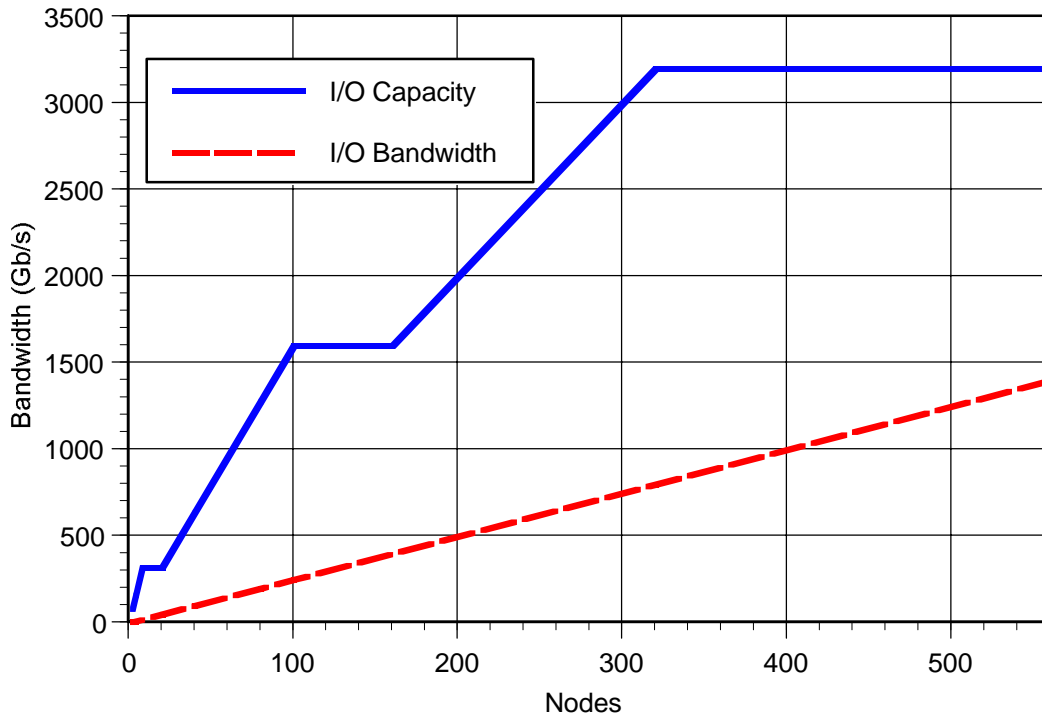


Figure 4: Bandwidth Scaling of the TSR

The TSR can be scaled economically because each line card carries its own portion of the active switching fabric. As the network is expanded, only the portion of the switching fabric needed by the actual number of nodes is present, and the cost of scaling is linear with the number of nodes.

In contrast, other network topologies either scale superlinearly or carry considerable unused switching capacity in small configurations. With a crossbar network, the entire switching fabric must be present to connect two line cards and the cost of scaling is quadratic. Butterfly networks can only be expanded in powers of their radix (usually 2) and the cost of scaling is superlinear. Each node of a binary n -cube (hypercube) network must be configured with ports for the maximum dimensionality of the network, even though these ports are unused in small configurations.

In addition to scaling economically, the TSR also scales *incrementally*. The torus network of the TSR may be expanded one line card at a time. While many of the networks along this scaling path are not regular tori, they are all completely connected (any node can reach any other node) and have a speedup of at least 4 up to 320 nodes. In addition, if the TSR is expanded two line cards at a time, all configurations with an even

number of nodes have at least two edge-disjoint paths between all pairs of nodes providing fault tolerance.

3.4. The path diversity of a torus network gives good load balance and high reliability

A large number of distinct paths exist between every pair of nodes in a torus network. By dividing traffic over these paths, load can be easily balanced across the network channels, even for very irregular traffic patterns. This path diversity also enables the network be quickly reconfigured around faulty channels, by routing traffic along alternative paths.

In a three-dimensional torus network, there are $|P| = \binom{\Delta x + \Delta y + \Delta z}{\Delta x} \binom{\Delta y + \Delta z}{\Delta y}$

distinct minimal-length paths from a node a to a node b whose coordinates differ by Δx , Δy , and Δz in the x -, y -, and z -dimensions respectively. An even larger number of paths is available if non-minimum length paths are allowed as well. For example, in an $8 \times 8 \times 8$ torus, an average message can choose between $|P| = 90$ 6-hop paths between its source and destination, and a maximum distance message, between two opposite points in the torus, can choose between 34,650 12-hop paths across the fabric. Distributing traffic across these large numbers of paths evenly balances the load on the fabric channels even when the traffic pattern is very non-uniform.

To see the importance of path diversity and load balance, it is instructive to examine what happens in a network without path diversity. Whenever there is sharing of channels but no path diversity, congestion due to load imbalance results on certain traffic patterns. In the 16-node radix-2 butterfly of Figure 5, as in all butterfly networks, there is exactly one path from each input to each output. For example, a packet travelling from node 0 to node 2 must travel through switches 00, 10, 20, and 31. There is no other way to get to node 2. Because of this inflexibility in routing, the butterfly is unable to balance load across its channels for non-uniform traffic patterns¹⁰.

The bold lines in Figure 5 show how congestion occurs when nodes 0,1,8, and 9 send packets to nodes 0,2,1, and 3. All of these packets must traverse the channel between

¹⁰ Non-uniform traffic patterns, where some input, A, is more likely to send traffic to one output, B, than some other output, C, are the norm in Internet routers. Uniform traffic patterns are almost never seen.

switches 10 and 20. Thus, the load on this channel becomes $\gamma_{100} = 4$ while the loads on the lower channel out of switch 10 and both channels out of switch 14 are zero, $\gamma_{101} = \gamma_{140} = \gamma_{141} = 0$. While the average channel load is 1, this load is not balanced across the channels. This load imbalance causes the throughput of this network to drop to 25% of its capacity on this traffic pattern. In general, the throughput of an N -node butterfly drops to $\frac{1}{\sqrt{N}}$ of its capacity on a worst-case traffic pattern. While the worst-case traffic pattern is fairly rare, traffic patterns that cause significant congestion due to load imbalance are quite common. Half of all permutation patterns¹¹ cause 2:1 congestion, and one quarter of all patterns cause 4:1 congestion.

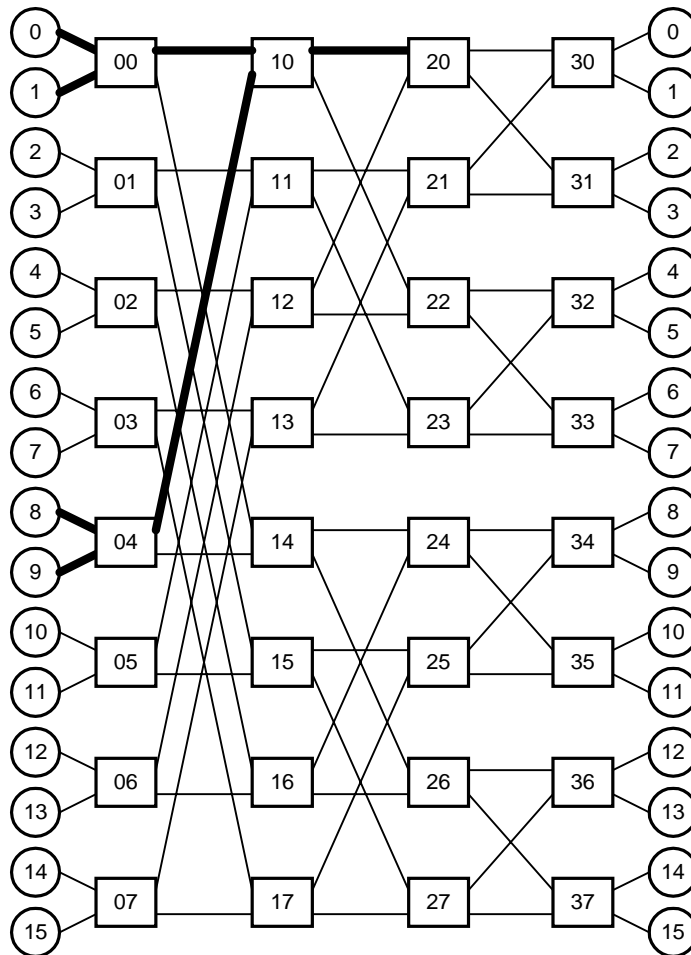


Figure 5: Congestion caused by load imbalance in a butterfly network

¹¹ A permutation traffic pattern is one in which each input node, A, sends all of its traffic to one output node, B.

In contrast, a torus network gives good load balance, and hence good performance, on all of these traffic patterns because it divides the load between each input and output across a large number of paths. Thus, even when the traffic pattern is unbalanced, the load on the channels remains balanced.

The high-path diversity of a torus network also enables the network to have high availability. If any channel or node goes down, the remainder of the network is able to continue operation by routing traffic around the failure. In an Avici TSR with an even number of nodes, there are at least two *edge disjoint* paths between every pair of nodes in the network. Two paths from node A to node B are edge-disjoint if they share no edges. For example, Figure 6 shows three edge-disjoint minimal (4 hop) paths from node A to node B. To provide alternative paths between nodes that are aligned with one another along an axis of the torus, the TSR allows limited non-minimal routing. Figure 6, for example shows two edge-disjoint paths from node C to node D which are aligned in the y-z plane. The solid blue path is minimal (2 hops) while the dashed red path is non-minimal (4 hops).

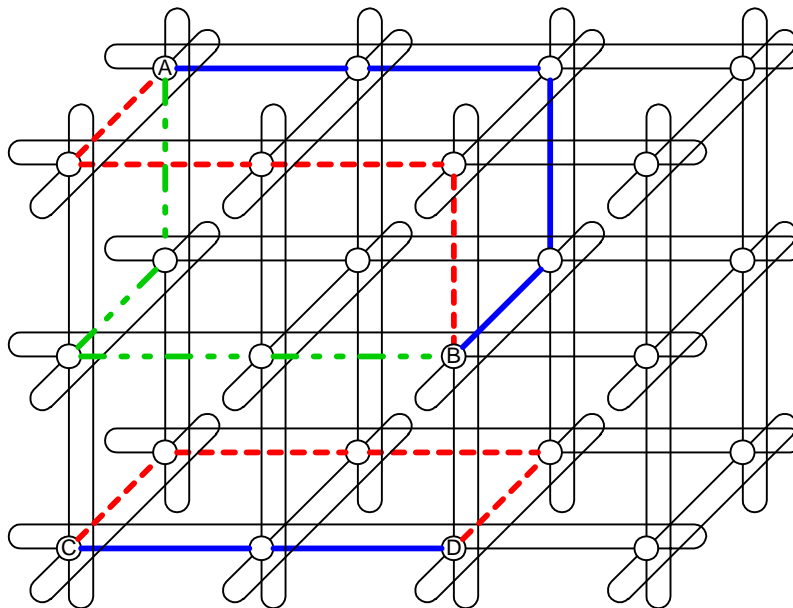


Figure 6: Minimal and Non-Minimal Edge-Disjoint Paths in a Torus

In addition to path diversity, a high-availability fabric also requires hardware support for fault detection, containment, reconfiguration, and repair. The Avici TSR constantly monitors the status of all fabric channels by periodically (every 100 μ s) performing a

channel CRC check. When a channel error is detected, the fault is contained by deactivating the channel and routing traffic on alternative paths that avoid the faulty channel. This rerouting is performed by *deflecting* packets for a short period until the fabric routing tables are reconfigured to avoid the faulty link. Finally, the TSR supports hot plugging of all active components to facilitate on-line repair.

3.5. Compared to other topologies, the torus network offers the most economical approach to scalable, extensible, reliable bandwidth

Table 1 lists four important properties of six popular network topologies. Each cell of the table is shaded to indicate suitability for use as a network fabric, a green cell is ideal for use as a network fabric, a yellow cell is problematic, and a red cell is completely unsuitable.

First, a network fabric that shares channels between destinations must have path diversity to provide load balance and fault tolerance. The formulae for the path diversity of each network are listed in Table 2 and the number is summarized in the first column of Table 1. Lack of path diversity rules out the butterfly topology. It has no path diversity and hence is subject to extreme load imbalance¹². The crossbar is problematic here as it offers no fault tolerance unless it is completely duplicated.

Table 1: Properties of Network Topologies

	Path Diversity	Bandwidth Scaling	Extensibility	Channel Length
2-D Torus	High	\sqrt{N}	1 Node	All Short
3-D Torus	High	$N^{\frac{1}{3}}$	1 Node	All Short
Hypercube	High	N	1 Node (limited by degree)	Some Long
Butterfly	None	N	Doubling	Some Long
Benes	High	N	Doubling	Some Long
Crossbar	N/A	N	None	Some Long

¹² The load-imbalance of the butterfly network can be remedied to some extent by adding extra stages to the network [AdamSieg82]. However, the load imbalance is not completely removed until every stage of the butterfly has been duplicated, resulting in a Benes network. The intermediate extra-stage butterflies would be colored yellow in this table.

Two columns of the table relate to scalability. All of the networks except the 2-D torus offer adequate bandwidth scaling. The 2-D torus is problematic in this respect because its bisection bandwidth only grows as the square root of the number of nodes. Thus this topology requires that the bandwidth of small systems be drastically over-provisioned so that adequate bandwidth remains when scaling to a maximal system.

The next column indicates the granularity at which the network can be extended. All three cube networks can be extended one node at a time. However the hypercube is problematic because this extensibility is limited by the degree of the node. Each node must be provided with the number of channels, $n = \lg(N)$, required by a maximal system even though these channels are unused in smaller systems. Thus, with this topology either scalability is limited, or small systems are prohibitively expensive. The butterfly and Benes networks are only extensible by powers of two, viz. by doubling the network, and the crossbar is not extensible at all. Thus, these three topologies are unsuitable for use as a scalable network fabric.

The final column of the table relates to economy rather than feasibility. Only the two low-dimensional networks have uniformly short connections between nodes. The other four networks require a substantial number of long channels making them significantly more costly to implement.

Of the six topologies considered, the three cube networks are suitable for use as network fabrics. Of these, only the 3-D torus is not problematic.

Table 2: Distance, Channel Load, Channel Width, and Path Diversity for Several Network Topologies

	D_{avg}	γ_{max}	W	Path Diversity
2-D Torus	$\frac{\sqrt{N}}{2}$	$\frac{\sqrt{N}}{8}$	$\frac{P}{4}$	$\left(\begin{array}{l} \sqrt{N}/2 \\ \sqrt{N}/4 \end{array} \right)$
3-D Torus	$\frac{3N^{\frac{1}{3}}}{4}$	$\frac{N^{\frac{1}{3}}}{8}$	$\frac{P}{6}$	$\left(\begin{array}{l} 3N^{\frac{1}{3}}/4 \\ N^{\frac{1}{3}}/2 \\ N^{\frac{1}{3}}/4 \end{array} \right)$
Hypercube	$\frac{\lg(N)}{2}$	0.5	$\frac{P}{2\lg(N)}$	$\left(\frac{\lg(N)}{2} \right)$
Butterfly	$\log_k(N)+1$	1	$\frac{P}{4}$	1

Benes	$2\log_2(N)+1$	1	$\frac{P}{4}$	N
Crossbar	2	1	$\frac{P}{2N}$	1

In addition to the qualitative measures listed in Table 1, a good topology must also scale economically, offering a low cost per node that grows slowly as the network is expanded. Figure 7 compares the cost per node of four popular topologies holding the bandwidth per node constant. Cost here is computed based on the number of chip, board, backplane, short cable, and long cable pins required with relative costs of 2, 1, 1, 4, and 16 respectively. Long cables, over 2m in length, require either optical signaling or repeaters to operate at full speed and thus are significantly more expensive than short cables (the 4x assumed here is conservative). Topologies where channels travel in both directions between a pair of nodes (Binary n-Cube and 3-Cube) are assumed to use simultaneous bidirectional signaling [DDL93] to share pins between channels traveling in opposite directions. This analysis assumes that 256 differential signals (512 signal pins) can be accommodated on each fabric router chip.

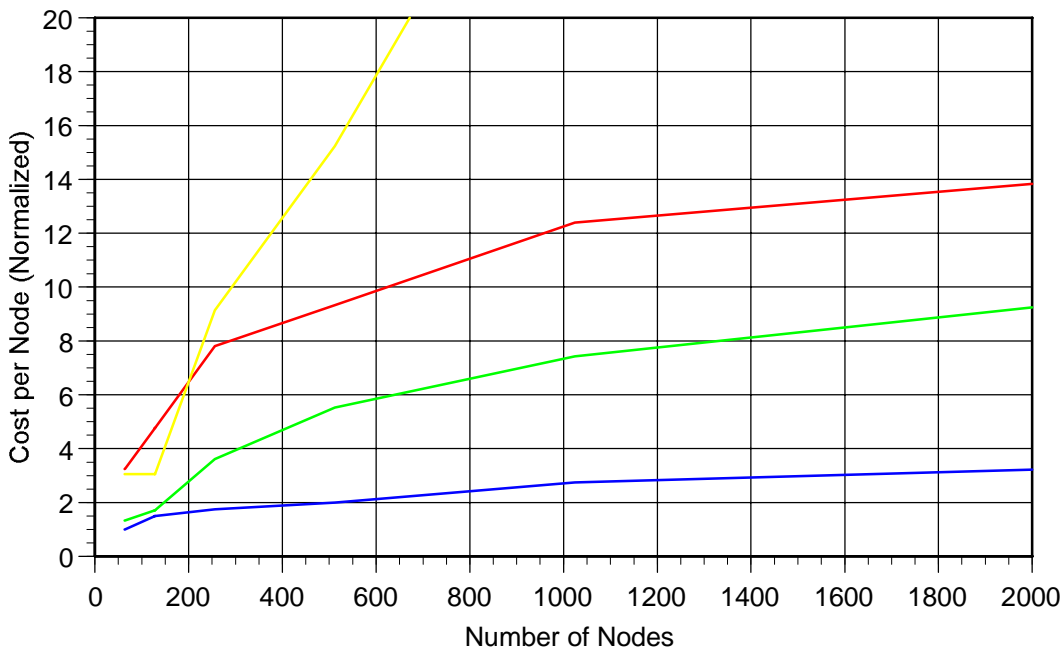


Figure 7: Relative Cost of Different Topologies

The figure shows that the 3-cube topology offers the lowest cost per node across the range of scaling. For small numbers of nodes (64 and 128), both cube topologies give lower cost for three reasons. First, by incorporating switching into each node, cabling to and from a separate switch component is eliminated. Second, because all channels are bidirectional, pins are shared between the channels in opposite directions. Finally, these topologies exploit locality to eliminate unneeded hops. For example, in an n -stage radix-2 butterfly, each packet must take $n+1$ hops to get to its destination. In binary n -cube, on the other hand, the average packet only takes $n/2$ hops since it is already in the right position in half of the dimensions.

As the number of nodes increases, the topologies scale in three ways. The crossbar scales quadratically, giving a linear increase in cost per node. This gives prohibitively high cost for more than 128 nodes. The butterfly and the binary n -cube fabrics scale as $N \log(N)$, but with a large constant factor due to the need for long cables with repeaters or optical transceivers. These networks are feasible but costly. Also, in addition to the cost of the components, they also present a burden in terms of increased power dissipation and the complexity of the required cabling. Finally, the 3-cube scales as $N^{\frac{2}{3}}$ with a small constant factor because most connections are made on backplanes with just a few short cables.

Some researchers have proposed switching fabrics based on multiple networks. Examples include the rotator network and a switched shared memory architecture [Chatter98]. As illustrated in Figure 8, these architectures consist of an input network, a shared memory, and an output network. Such dual-network fabric architectures incur twice the cost of a single network fabric because they include two separate networks, each of which must have the capacity to handle the full input bandwidth. In addition to doubling the cost, these dual-network architectures degrade quality of service. Packets metered out of the shared memory encounter the variable delay of the output network before arriving at the output fiber. Thus, the QOS algorithms at the shared memory do not have complete control over the departure time of each packet. For reasons of both cost and quality of service, it is far better to use a single network fabric architecture that acts as a *virtual crossbar* and to queue packets at the output port.

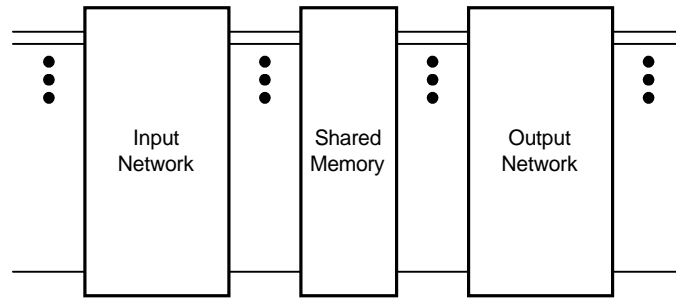


Figure 8: Dual-Network Fabric Architecture

The path diversity, incremental scalability, and locality of the 3-D torus network make it uniquely well suited to the needs of scalable switching fabrics. These properties have made the 3-D torus the network of choice for massively parallel computers. The same properties make the 3-D torus the network of choice for scalable routers.

4. Virtual networks make the torus look like a *virtual crossbar*

To provide the quality of service required by an Internet router, the Avici TSR fabric provides two separate virtual networks with completely separate buffers for each output port of the network (a total of 1120 virtual networks in a 560-port system). One set of virtual networks serves best-efforts traffic while the other set supports guaranteed-bit-rate traffic. The virtual networks prevent contention between packets of different service classes and packets destined for different output ports. In effect, the torus with virtual networks acts as an output-queued virtual crossbar: providing non-blocking³ service from any idle input to any idle output, and providing low, bounded delay for guaranteed-bit-rate traffic. Packets to outputs that are not oversubscribed are delivered across the fabric with minimum delay and queued at the output where they are metered onto the outgoing line according to the QOS policy.

4.1. *Virtual networks prevent tree saturation making the network non-blocking*

In an Internet router, it is possible for a single output port to be oversubscribed by best-efforts traffic for short periods of time. Without virtual networks, packets destined for the hot-spot node (oversubscribed output port) would consume all available fabric buffers along paths to this node. With the supply of buffers along these paths exhausted, packets destined for unrelated output ports are not able to make progress and block

themselves, making even more buffers unavailable. The blockage rapidly spreads until the entire network is congested. This cascaded blocking of channels is called *tree saturation* [PfisNort85] because the congestion forms in a tree structure rooted at the hot-spot node. When a network is tree saturated, throughput is reduced to a small fraction of its original capacity.

Tree saturation occurs due to contention for buffers, not for channel bandwidth. The hot-spot node can only remove packets from the network at a limited rate. Because this load is evenly distributed across network channels, the hot-spot node places only a modest bandwidth demand on each channel. For example, in the Avici TSR the extraction bandwidth is limited to twice the channel rate (eight times the line rate). This bandwidth of $2b_c$ is divided across the six channels into the hot node, loading each to $b_c/3$ with hot-spot traffic. The twelve channels feeding these channels are in turn loaded to $b_c/6$ with hot-spot traffic, and so on. The hot-spot traffic uses no more than $1/3$ of the available bandwidth of any channel. This leaves plenty of bandwidth to handle traffic destined for other output ports if buffers are available to stage the flits of these packets.

The TSR guarantees buffer availability by providing a separate set of flit buffers on each physical channel for each output port and class of service. Each of these buffer sets acts as a virtual channel [Dally92]. The set of virtual channels associated with a given output forms a virtual network, a tree of virtual channels rooted at the output node. This dedicated virtual network enables packets to be forwarded from any input to the output at the root of the virtual network without competing for buffers with traffic destined for different outputs. Thus, tree saturation is not possible in a fabric with virtual networks, and packets are delivered to outputs that are not oversubscribed in a non-blocking³ manner.

Packets do share physical link bandwidth with packets from other virtual networks. However, the contention for this bandwidth is negligible. Because of the load-balance and speedup of the network, a physical channel is never loaded to more than $2/3$ of its capacity, even in the presence of a nearby hot-spot. With the physical channels multiplexed over competing packets in units of 72-Byte flow-control digits or flits, the worst-case expected waiting time to access a link is 60ns per hop, or 360ns for a typical 6-hop route, a negligible amount.

With virtual networks and good load balance, the torus network of the Avici TSR provides non-blocking³ packet delivery. That is, packets destined for one output are not substantially delayed due to contention with packets destined for a different output¹³. The load balance limits the peak load on a physical channel making contention for physical channels bandwidth negligible, and the virtual networks eliminate contention for buffers.

4.2. Routing guaranteed bit-rate traffic on a separate set of virtual networks guarantees a low, bounded delay for this traffic

The Avici TSR provides a separate set of virtual networks for guaranteed-bit-rate (GBR) traffic and gives GBR traffic priority over best-efforts (BE) traffic in competing for link bandwidth. These policies ensure that GBR traffic will see a low, bounded delay when traversing the network.

Consider a GBR packet that must traverse the longest path (14-hops) in a 560-node, $14 \times 8 \times 5$, torus. In the absence of contention, this traversal takes 310ns per hop or $4.4\mu\text{s}$ from end to end. Contention for channel bandwidth is negligible¹⁴, and buffer contention is limited to other GBR packets destined for the same output. In the (very unlikely) worst case, 560 long (1KByte) GBR packets destined for a single output, A, arrive at the 560-ports of the network simultaneously. These 560 packets will be transported by the fabric to port A at a rate limited by the extraction channel at port A. At the extraction bandwidth of 20Gb/s (twice the channel rate or 8 times the line rate), it takes $28\mu\text{s}$ for node A to clear these 560KBytes worth of packets from the network. Because the GBR traffic is guaranteed not to oversubscribe the output port, another packet for this output will not arrive at any input until $224\mu\text{s}$ after the arrival of the first wave, allowing plenty of time for the fabric to be cleared of the first wave. Thus, even in this worst-case situation, no GBR packet sees more than $33\mu\text{s}$ of delay ($28 + 4.4$) in traversing the TSR fabric¹⁵.

¹³ In fact, because the extraction bandwidth of the torus is many times the line rate, the TSR can handle traffic from two inputs to one output without contention. This is something that a circuit-switched crossbar cannot do.

¹⁴ Physical channel contention for GBR traffic is even smaller than for BE traffic because the GBR traffic is guaranteed not to be oversubscribed, thus there are no hot-spots in the GBR network.

¹⁵ Of course in this worst-case situation, at least one of the GBR packets will be delayed up to $224\mu\text{s}$ in the output queue. However, this would be the case regardless of the fabric used.

5. Conclusion

The 3-D torus topology with virtual networks is uniquely well suited to serve as a scalable switching fabric for the next generation of Internet backbone routers. The torus topology is economically scalable from a single node to thousands of nodes and can be extended or upgraded a single node at a time. The high path diversity of the network provides high availability and in combination with a randomized routing strategy balances load across the network channels allowing the fabric to handle non-uniform traffic loads without performance degradation. The torus fabric can be realized using only short, bidirectional channels giving significant advantages in cost, power, and complexity of cabling compared to topologies that require long wires.

Virtual networks, along with the load balance from path diversity make the torus behave as an output-queued crossbar switch. Providing a separate virtual network, a network of virtual channels, for each output port eliminates buffer contention between traffic destined for different outputs making the torus network non-blocking³. Constant-bit-rate traffic is guaranteed a bounded, low delay through the fabric ($< 33\mu\text{s}$) by providing a completely separate set of virtual networks for GBR/CBR traffic and giving this traffic priority over best-efforts traffic in allocation of fabric channel bandwidth.

Other fabric architectures fall short in providing scalable performance with quality of service. Bus-based systems are not scalable and crossbars are prohibitively costly beyond a small number of nodes. Butterfly (also called Banyon or shuffle-exchange) networks have no path diversity and hence suffer congestion on non-uniform traffic patterns and reduced availability. Butterflies, Benes networks, and crossbars are not incrementally extensible. They can only be expanded in large units. Butterflies, Benes networks, binary n -cubes (hypercubes), and crossbars all require long wires increasing cost, power, and complexity. Without virtual networks, all of these architectures, except crossbars, are subject to tree congestion and hence packet blocking, and have difficulty providing the quality-of-service guarantees required by critical traffic. Dual-network architectures with a central shared memory suffer from the deficiencies of their constituent network architectures as well as the increased cost of doubling the network and imprecise quality-of-service due to midpoint queuing.

The 3-D torus with virtual networks uniquely provides the combination of economy, scalability, and quality of service needed for tomorrow's Internet routers.

6. References

- [AdamSieg82] G. Adams, and H. Siegel, "The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems" *IEEE Transactions on Computers*, C-31(5), pp. 443-454, May 1982.
- [Agarwal91] A. Agarwal, "Limits on Interconnection Network Performance," *IEEE Transactions on Parallel and Distributed Systems*, PDS-2(4), pp. 398-412, October 1991.
- [Carbonaro96] J. Carbonaro and F. Verhoorn, "Cavallino: The Teraflops Router and NIC," *Proc. Hot Interconnects IV*, 1996.
- [Chatter98] M. Chatter, "Multi-port internally cached DRAM system utilizing independent serial interfaces and buffers arbitrarily connected under a dynamic configuration," U.S. Patent 5,799,209.
- [Dally90a] W. J. Dally, "Network and Processor Architectures for Message-Driven Computing," in *VLSI and Parallel Computation*, Suaya and Birtwistle Eds., Morgan Kaufman, 1990.
- [Dally90b] W. J. Dally, "Performance Analysis of k -ary n -cube Interconnection Networks," *IEEE Transactions on Computers*, C-39(6), pp. 775-785, June 1990.
- [Dally91] W. J. Dally, "Express Cubes: Improving the Performance of k -ary n -cube Interconnection Networks", *IEEE Transactions on Computers*, C-40(9), pp. 1016-1023, September, 1991.
- [Dally92] W. J. Dally, "Virtual Channel Flow Control," *IEEE Transactions on Parallel and Distributed Systems*, PDS-3(2) pp. 194-205, March 1992.
- [DLD93] L. Dennison, W. Lee and W. Dally, "High Performance Bidirectional Signalling in VLSI Systems," *Proceedings of the Symposium on Integrated Systems*, MIT Press, March 1993, pp. 301-319.
- [DLAPT98] W. J. Dally, M-J. E. Lee, F-T. An, J. Poulton, and S. Tell, "High-Performance Electrical Signaling," *Proc. Fifth International Conference on Massively Parallel Processing using Optical Interconnects*, 1998, pp. 11-16.
- [DYN97] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks, an Engineering Approach*, IEEE Computer Society Press, 1997.
- [KessSchw93] R. E. Kessler and J. L. Schwartzmeir, "Cray T3D: A New Dimension for Cray Research," *Compcon*, pp. 176-182, Spring 1993.
- [PfisNort85] G. Pfister and A. Norton, "Hot-Spot Contention and Combining in Multistage Interconnect Networks," *IEEE Transactions on Computers*, C-34(10), pp. 943-948, October 1985.
- [ScotThor94] S. L. Scott and G. Thorson, "Optimized Routing in the Cray T3D," *Proc. Workshop on Parallel Computing Routing and Communication*, pp. 281-294.
- [ScotThor96] S. L. Scott and G. Thorson, "The Cray T3E Network: Adaptive Routing in a High-Performance 3D Torus," *Proc. Hot Interconnects IV*, 1996.