

The Data Vortex, an All Optical Path Multicomputer Interconnection Network

Cory Hawkins, *Student Member, IEEE*, Benjamin A. Small, *Member, IEEE*,
D. Scott Wills, *Senior Member, IEEE*, and Keren Bergman, *Member, IEEE*

Abstract—All optical path interconnection networks employing dense wavelength division multiplexing can provide vast improvements in supercomputer performance. However, the lack of efficient optical buffering requires investigation of new topologies and routing techniques. This paper introduces and evaluates the Data Vortex optical switching architecture which uses cylindrical routing paths as a packet buffering alternative. In addition, the impact of the number of angles on the overall network performance is studied through simulation. Using optimal topology configurations, the Data Vortex is compared to two existing switching architectures—butterfly and omega networks. The three networks are compared in terms of throughput, accepted traffic ratio, and average packet latency. The Data Vortex is shown to exhibit comparable latency and a higher acceptance rate (2x at 50 percent load) than the butterfly and omega topologies.

Index Terms—Optical switch fabrics, optical switching, photonic packet switch, data vortex switch architecture, packet switching.

1 INTRODUCTION

SUPERCOMPUTERS harness a concurrent organization of the highest performance processing technology to deliver superlative performance. Commercial microprocessors that benefit from amortized design cost and economies of scale are an attractive candidate for future supercomputers, especially as their available off-chip bandwidth is increased. Commercial memories provide the only viable option for large-scale supercomputer storage. However, analogous commercial interconnection network technology lacks the necessary performance (latency and throughput) to satisfy supercomputing needs.

According to the TOP500 Supercomputer Sites website, all of the current top 25 supercomputers in the world have more than one thousand processors, and the top three have tens of thousands of processors [30]. With the trend of increasing processor count to achieve greater system performance, more pressure is placed on the performance of the interconnection network. The combination of increasing parallelism and clocks rates is pushing wire networks to their throughput limits. At the same time, applications and programming environments benefit from high throughput, low latency networks in order to balance computational load and exploit concurrency. The ideal network offers the following features:

- access latency on the order of the time-of-flight for both large and small data transfer,
 - necessary throughput to support high traffic volume with minimal delay,
 - efficient hardware and software interfaces to commercial processors and memory, and
 - scalability to thousands of nodes.
- C. Hawkins and D.S. Wills are with the Department of Electrical Engineering, Georgia Institute of Technology, 777 Atlantic Drive NW, Atlanta, GA 30332-0250. E-mail: {cory, scott.wills}@ece.gatech.edu.
- B.A. Small and K. Bergman are with the Electrical Engineering Department, Columbia University, 500 West 120th Street, Room 1300, New York, NY 1027-4712. E-mail: {bas, bergman}@ee.columbia.edu.

Manuscript received 12 July 2005; revised 21 Dec. 2005; accepted 11 Apr. 2006; published online 25 Jan. 2007.

Recommended for acceptance by C. Raghavendra.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0335-0705.

- necessary throughput to support high traffic volume with minimal delay,
- efficient hardware and software interfaces to commercial processors and memory, and
- scalability to thousands of nodes.

Optical technology offers the opportunity for transporting unprecedented data bandwidths across the interconnection computing network. Dense wavelength division multiplexing (DWDM) enables bandwidth in the TByte/s range ($40 \text{ Gbit/s} \times 250 = 10 \text{ Tbit/s}$) in each optical fiber interconnection port and for a 10,000 node system the throughput (bisectional) bandwidth can approach 10 PByte/s. An optical packet interconnection network can capitalize on this enormous bandwidth advantage and additionally deliver minimum latency across the extent of a large-scale supercomputer (100m). Achieving this requires a truly packet routing, optical switching fabric where congestion is locally resolved. One of the key challenges, however, to implementing viable large-scale optical packet switching fabrics has been the lack of adequate random-access optical buffering; multiple-wavelength data structures which utilize DWDM present an additional challenge due to the large optical bandwidth required [5]. Optical buffers are generally designed to work for a single wavelength so when a multi-wavelength packet needs to be buffered it needs to be split up into multiwavelength parts, and each individual channel is buffered (so N buffers are needed for N-channels). Interconnection network architectures considered for implementation in the optical domain should require minimal or no packet buffering and avoid complex routing data processing. Self-routing deflection network topologies are the most likely candidates for optical packet switching fabrics. Due to the decreasing costs of optical components, some electrical network designs of multistage interconnection networks (MIN) are currently being redesigned for optical implementation [2], and some totally new optical network designs are being explored as well.

One such new network topology is the Data Vortex—a highly scalable optical packet switching architecture that utilizes self-routing of individual packets and alleviates the need for central scheduling and processing [3], [4]. Deflection routing is used to eliminate internal packet buffering and minimize packet traffic congestion, and the Data Vortex topology is designed especially to work in conjunction with deflection routing. Deflection routing is similar to hot-potato routing, in that when contention for a link occurs, one message is routed correctly, and the losing contender is routed along a different path (possibly in a direction farther away from the desired message destination) and has been used extensively in the past in optical interconnection networks [32], [33]. Deflection routing affords the omission of buffers at each node by taking advantage of the Data Vortex schematic design which allows an always-open path for each packet in the event of contention. The architecture's unique absence of internal optical buffering elements enables the transparent routing of DWDM packet payloads while maintaining flexibility for extending the packet size by simply adding (or removing) wavelength channels.

1.1 Data Vortex Architecture

The Data Vortex optical packet switching network architecture was designed specifically for realization in the optical domain, taking into consideration the difficulty of implementing optical buffering and complex optical logic [5], [6]. The architecture is a fully-implemented directed deflection routing topology composed of simple 2×2 switching elements, or nodes. The nodes are arranged in hierarchical levels or cylinders, each affixing an additional bit in the packet's destination address, in a manner similar to Banyan network addressing [2]. It is designed to facilitate optical implementation by maintaining simple routing and eliminating the need for internal physical buffering. The Data Vortex is an input-blocking architecture that exhibits no internal blocking and no output blocking. Contention within the network is resolved by simple deflection routing techniques. Deflection routing removes the need for buffers by allowing packet contention to be resolved without blocking within the network and without blocking at the output. Optical to electrical (O/E) and electrical to optical (E/O) signal conversion is not necessary since buffering is avoided altogether. This significantly reduces the overall cost of the network, as the necessary hardware is eliminated [5]. Additional benefits including a decrease in the operating power dissipation, a reduction in complexity, and an increase of switching speed are also realized.

The Data Vortex architecture has been simulated in previous work [6] to obtain preliminary performance data. Likewise, the Data Vortex switching nodes have been tested for proper function and routing [7], [26]. Metrics for performance measurements are chosen as latency in time of flight and packet acceptance as a percentage of offered traffic. These two metrics allow an overall view of network throughput and latency for comparison to existing topologies. Even though previous work has measured performance, the Data Vortex has not yet been compared to existing architectures in order to evaluate its relative latency and packet acceptance under similar workloads. This is the goal of the simulations illustrated in this paper.

1.2 Data Vortex Implementation

The Data Vortex architecture has been implemented on numerous occasions with contemporary fiber optic components in a laboratory setting [7], [8], [9], [10], [11], [23], [24], [25], [26]. It is envisioned that similar components would be used in a commercial implementation of the switching fabric. The switching elements themselves are constructed from silica fiber optic couplers and filters, semiconductor optical amplifiers (SOAs), solid state optoelectronic receivers, and some simple high-speed electronics for the execution of the routing decision (*ibid.*). The optical components are capable of working over a bandwidth of about 6 THz near 1,550 nm (the ITU C-band), allowing for packets with extremely large data rates (i.e., terabits per second [12]) to be transmitted simultaneously through the same components and optical fiber. Packets are composed of many independently modulated channels (wavelengths) transmitted in parallel; particular channels are used for routing headers, and others are designated for the payload [26], [27].

2 DEFINITION OF NETWORK TOPOLOGY

First proposed in 1998 [3], the Data Vortex architecture incorporates both deflection routing and Banyan-style hierarchal addressing while utilizing bufferless switching nodes. This is accomplished by extending the paths in a conventional butterfly network to allow for routes which are always available for packet deflection. The topology of Data Vortex networks are thus quite intricate and are best visualized in three dimensions (Fig. 1). Because the packet paths are arranged angularly in this three-dimensional schematic with traffic generally flowing inward, the aggregate flow of traffic resembles a physical spiral or vortex.

2.1 Physical Topology

The Data Vortex topology is composed entirely of 2×2 switching elements (also called switching nodes) arranged in a fully connected, directed graph with terminal symmetry, but not complete vertex symmetry. The single-packet routing nodes are wholly distributed and require no centralized arbitration, and function only as cross-points for routing one packet at a time to one of two outputs without any buffering or state information [14], [27]. The topology is divided into C hierarchies or cylinders which are analogous to the stages in a conventional banyan network (e.g., butterfly). The architecture also incorporates deflection routing, which is implemented at every node; deflection signal paths are placed only between different cylinders. Each cylinder (or stage) contains A nodes around its circumference and $H = 2C - 1$ nodes down its length. The topology contains a total of $N = AHC = AH(\log_2 H + 1)$ switching elements, with $N_i = AH$ possible input terminal nodes and the same number of possible output terminal nodes, arranged on the outer and inner cylinders, respectively. The position of each node is conventionally given by the discrete triplet (a,c,h) ,

$$0 \leq a \leq A - 1, 0 \leq c \leq C - 1, 0 \leq h \leq H - 1.$$

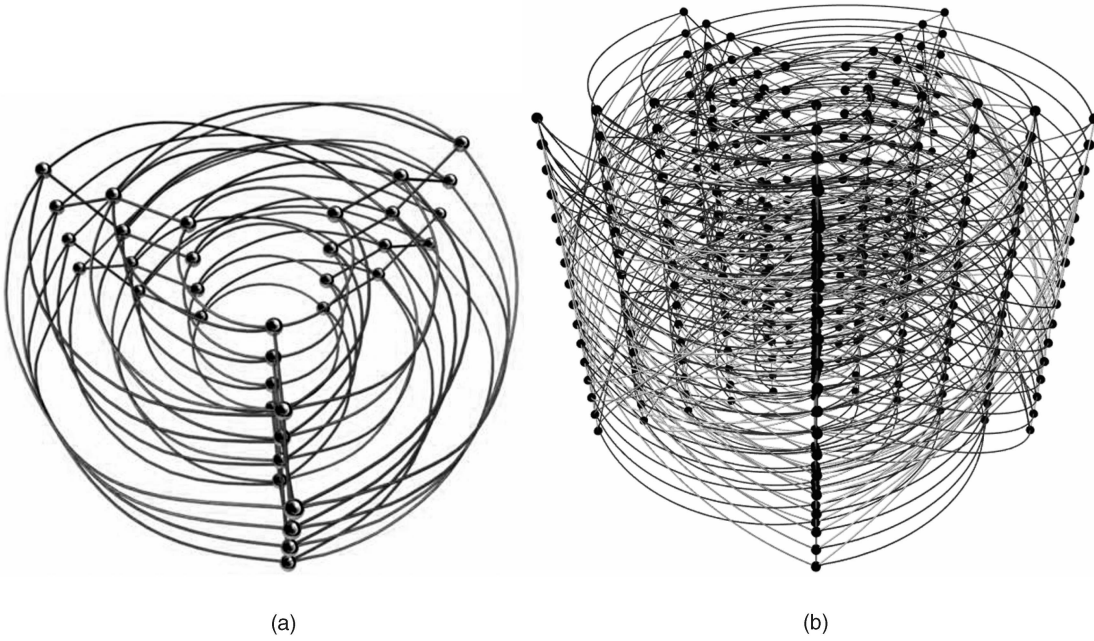


Fig. 1. (a) Illustration of an example Data Vortex topology with three angles (A), a total height of four (H), and three cylinders (C). (b) A second example of a Data Vortex topology with $A = 5$, $H = 16$, and $C = 5$.

Packets are injected only on designated slot times, and each packet must be wholly contained within the slot. Furthermore, the duration of the slots is set to match the propagation delay from each node to an immediately adjacent one [26], [27]. This slotted timing allows for an exceedingly simple node structure which does not require buffering. Because the Data Vortex is meant to be used as an interconnection network in closed systems (e.g., multiprocessor supercomputers), this injection constraint can be maintained by using an appropriate terminal clock distribution scheme [28].

Paths within a cylinder exist only between nodes of adjacent angle values and never between nodes with the same position around the circumference of the cylinder; i.e., only from (a, c, h) to $(\text{mod}A \ a + 1, c, G_c(h))$, where $G_c(h)$ is a mapping which defines the “crossing pattern” of the height coordinate between adjacent angles for cylinder c [4], [26]. These edges are often termed deflection paths because, while they are also used for address resolution, they are the only links available for deflections. Additional edges are present between cylinders called ingress paths, which connect nodes of the same height and of adjacent angle values; i.e., from (a, c, h) to $(\text{mod}A \ a + 1, c + 1, h)$. Thus, all paths between nodes progress one angle dimension forward and either continue around the same cylinder while moving to a different height, or ingress to the next hierarchical cylinder at the same height. Deflection signals connect only nodes on adjacent cylinders with the same angular dimension; i.e., from $(a, c + 1, h)$ to a node at position $(a, c, G_c + 1(h))$. The conventional nomenclature illustrates packets routing to progressively higher numbered cylinders as moving inward toward the network outputs.

The paths within a cylinder differ depending upon the level c of the cylinder. The crossing or sorting pattern (i.e., the connections between height values defined by $G_c(h)$) of the outermost cylinder ($c = 0$) must guarantee that all paths

cross from the upper half of the cylinder to the lower half of the cylinder so that the graph of the topology remains fully connected, and so that the Banyan-like bitwise addressing scheme functions properly. Inner cylinders must also be divided into $2c$ fully connected (viz., Hamiltonian) and distinct subgraphs, depending upon the cylinder. Only the final level or cylinder ($c = C - 1$) may contain connections between nodes of the same height. The cylindrical crossing must ensure that destinations can be addressed in a binary tree-like configuration, similar to binary banyan networks.

Packets can be injected at all angles along the input or only at a fraction of the angles, as determined by a chosen asymmetry ratio (A'/A , where A' is the number of angles that packets are potentially injected upon) [6]. The angle value can be viewed as the amount of “virtual buffering” for the network, as data propagates forward one angle on each cycle. If deflections that keep the packets in the same cylinder are necessary, adding more angles increases the number of packets each cylinder can accommodate, thus increasing the level of virtual buffering within the network. In order to optimize fairness and to minimize the likelihood of starvation, investigations have shown that the angle number should be a small, odd integer (e.g., 3, 5, 7), and that the cylindrical paths should symmetrically follow the crossing patterns shown in Figs. 1 and 2, which guarantee a full sorting of height values [13]. The G_c incidence matrices can easily be found by calculating the real non-negative $2C - c - 1$ th root or roots of the $2C - c - 1$ identity matrix and, thus, the graph spectra are the $2C - c - 1$ th unity roots. These crossing patterns guarantee a full Hamiltonian tour of the cylinder for odd angle values, and the largest possible Hamiltonian tour, in general [13].

2.2 Node Addressing

Addressing within the Data Vortex architecture is entirely distributed and bitwise, similar to Banyan architectures: as

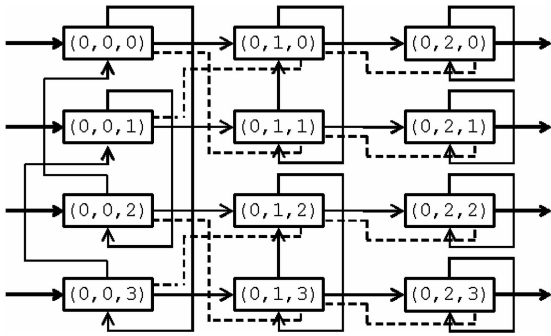


Fig. 2. Schematic of an $A = 1$, $C = 3$, and $H = 4$ Data Vortex network. Directed optical paths are shown as solid arrows, and electronic deflection signals as dashed lines. The 12 nodes are labeled with conventional coordinates (a, c, h) .

a packet progresses inward, each successive bit of the binary address is matched to the destination. Each cylinder tests only one bit (except for the innermost one), and half of the height values permit ingress for "1" values, and half for "0" values, arranged in a Banyan-like binary tree configuration. Within a cylinder c , nodes at all angles at a particular height (i.e., (\bullet, c, h)) match the same $c + 1$ st significant bit value while paths guarantee preservation of the c most significant address bits. Thus, with each ingress to a successive cylinder, progressively more precision is guaranteed in the destination address. Finally, on the last cylinder $c = C - 1$, each node in the angular dimension is assigned a least significant value in the destination address so that the packets circulate within cylinder $c = C - 1$ until a match is found for the last $\lceil \log_2 A \rceil$ bits (so-called angle-resolution addressing).

Each switching element within the interconnection network is bufferless and is designed to check exactly one bit of the destination address, in addition to a general presence bit (or packet frame). When the selected address bit matches the value assigned to the node because of its position with the cylinder, the packet is allowed to ingress into the next cylinder, unless a deflection signal is received. When the selected address bit does not match, or when a deflection signal is received, the packet is routed within the same cylinder, and the node sends a deflection signal indicating that the next node will soon be busy with that packet. Therefore, every switching element always has an available deflection path, and has an ingress path used for routing matches.

While it may seem wasteful to have twice as many optical paths as necessary, having a guaranteed deflection path allows for an extraordinarily simple routing logic which can be executed extremely quickly [14], [27]. The distributed bitwise addressing scheme also helps ensure that routing decisions do not dominate network latency. No buffers are used, so the network latency can be reduced to the optical time-of-flight.

2.3 Deflection Implementation

The nodes within the Data Vortex architecture are made to be as simple as possible. Although they are switching elements with in-degree and out-degree 2, these nodes are designed to route only one packet per time slot (see 1×2 switch) [14], [27].

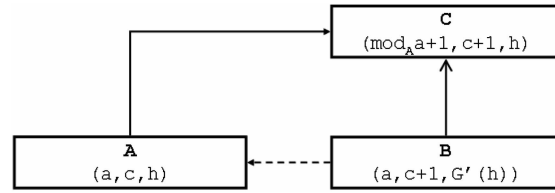


Fig. 3. Schematic of the deflection routing structure. Nodes $A(a, c, h)$ and $B(a, c + 1, G'(h))$ may route packets to node $C(\text{mod}_A a + 1, c + 1, h)$, where G' is the inverse transform of G . Whenever a packet from B is routed or deflected to C , node B sends a deflection signal to A , preventing an ingress to C . The packet from A must then reattempt ingress at a later time slot.

This fundamental constraint allows for a simplistic construction of the node but requires an architectural implementation of internal blocking or deflection. Deflection within the Data Vortex is thus a bit different from conventional deflection routing topologies which allow for deflection routing at the completion of each discrete node hop. Instead, the Data Vortex deflection implementation prevents two packets from ever entering the same switching element by controlling or blocking one of the two possible source nodes [27].

Thus, deflections occur only when a packet in cylinder $c + 1$ is deflected or remains in cylinder $c + 1$ due to its destination address. A packet which would otherwise ingress into that node on cylinder $c + 1$ from its node on cylinder c is thus required to remain in cylinder c , since the deflection signal indicates that the desired node will be busy (Fig. 3). This deflection structure results in a "back-pressure" from the inner cylinders ($c = C - 1, C - 2, \dots$) to the outer cylinders ($c = 0, 1, \dots$). A deflected packet must traverse two additional node hops before the address again matches, as a consequence of the crossing patterns' sorting between "0" and "1" addresses.

The deflection signaling structure continues to the nodes on the output and input terminals such that output nodes can receive busy signals from the output queuing subsystem, and the input nodes can transmit similar busy signals to the input interface. Thus, packets which attempt input at the first cylinder $c = 0$ may receive a signal which indicates that the desired input node is busy; the packet must therefore be queued to reattempt injection, or be discarded.

The deflection signal relationship can be represented geometrically as a triangle (as in Figs. 3 and 4) and necessarily implies the existence of similar triangles within the topological graph. In fact, this triangular deflection unit is the fundamental building block of the entire Data Vortex topology [27]. Only the arrangement of each leg differs from cylinder to cylinder, in accordance with the specific crossing patterns used. The deflection paths' crossing pattern must be the same as cylinder $c + 1$'s (i.e., $Gc + 1$) since connections between cylinders do not undergo height translation.

In order to maintain correct timing in the deflection signaling, particular latency conditions must be met. Because implementations of this architecture avoid buffering within the switching elements, latencies are caused entirely by the optical and electrical (for control signals only) paths' times-of-flight. To maintain accurate deflection signaling, deflection signals must be transmitted early enough so that the node

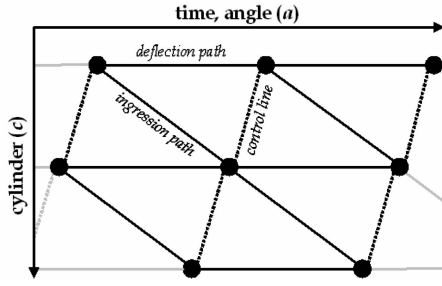


Fig. 4. Graphical depiction of the triangular relationship between the latencies of the deflection paths, ingress paths, and deflection lines. As shown, the latency of the deflection paths must be equal to the sum of the latencies of ingress and deflection signal transmission and processing. The vertical dimension simply represents ingress through the network, independent of time, while the horizontal dimension can be viewed as the progression or of either angle or time, since packets continually advance in the angular dimension.

receiving the deflection signal can direct its packet appropriately. The timing of cylinder $c + 1$ which contains the node initiating the deflection must therefore precede the timing of the cylinder c which contains the node receiving the deflection signal. Thus, the ingress paths must be shorter than the deflection paths by an amount equal to the processing and transmission time of the deflection signal. Consequently, the timing cycles of inner cylinders precess those of the outer ones (Fig. 4).

When the aforementioned timing condition is met for a Data Vortex implementation, global clocking for every switching element is not required. The precision with which this timing equality must be met has been investigated empirically in great detail, and is found to be reasonable for even the coarsest of fabrication techniques [28]. If packets are injected only at time slots which correspond to the deflection path latency, packets will maintain this slotted arrival schedule at every position within the hierarchical topology. Again, recall that no buffers or storage devices are used, so when physical time-of-flight requirements are met, they hold for all packets at each node.

2.4 Node Logic

In order to accomplish the aforementioned routing and deflection functionality, an individual switching node must process the following pseudocode routing logic, given its coordinates (a, c, h) :

```
function route (packet)
if (c < C - 1) then
  k ← 2C - c - 1
  b ← modkh ≥ k/2
  if (¬ deflection_in && packet.header(c) == b) then
    deflection_out ← false
    return (modA a + 1, c + 1, h)
  else
    deflection_out ← true
    return (modA a + 1, c, Gc(h))
end if
else // angle-resolution routing
  if (¬ deflection_in && packet.header(C - 1) == a) then
    // this last vector element may actually contain
    multiple
```

```
// bits to account for angle-resolution address
encoding
deflection_out ← false
exit
else
  deflection_out ← true
  return (modA a + 1, C - 1, h)
end if
end if
```

Practically, nodes are simply physically encoded with their b values or a values, regardless of c and h . In network implementations, the header information is actually transmitted optically, simultaneously with the packet payload, using a unique multiple-wavelength bit-parallel encoding scheme [14], [25], [26], [27]. This allows header bits to be separated from the rest of the packet with conventional fiberoptic wavelength filters for detection and conversion to electrical signals which can be used for the routing logic. This process does not alter the packet payload and maintains its high bandwidth.

2.5 Angle Considerations

Numerous operating schemes have been envisioned for the Data Vortex topology. The most common variations consider either reducing the number of input ports to H , so that only a single angle value $a' = 0$ is used for injection; or allowing the output ports on the same height h to be degenerate, eliminating the need for complex angle-resolution addressing at the innermost cylinder $c = C - 1$. The details and the trade-offs between these operating schemes is beyond the scope of the current introductory discussion, as is their relationship to the angle count A . We consider here only all-angle and single-angle injection without presenting a throughput, latency, nor cost analysis of the circumstances for which each mode of operation should be used. In either case, the fundamental structure of the architecture and of the switching elements is unaltered.

The total number of angles A which optimizes throughput and latency also depends on the injection scheme, in addition to the network size and implementation. Generally, because the total switching element count N is of higher order on H than it is on A (*viz.*, $N = AH(\log_2 H + 1)$), it is more cost-effective to increase the angle count instead of the height of the topology in order to accommodate a given number of terminal ports. However, when angle-resolution addressing is used, a system with too many angles experiences substantial backlog on the innermost cylinder (and, hence, “backpressure”). This topic also requires a more detailed discussion, which will be presented elsewhere.

Historically, in implemented systems and subsystems [6], [7], [8], [9], [10], [11], [23], [24], [25], [26], all angles have been used for injection, and angle-resolution routing has been used. However, the size and cost of these laboratory systems made increasing A more feasible than increasing H . Past simulations have considered both injection and addressing approaches [6], [15], and the current work extensively discusses single-angle injection Data Vortex networks.

2.6 Deadlock and Livelock

Regardless of the operating scheme, angle count, or traffic patterns used, the Data Vortex architecture is categorically deadlock-free, since it is a properly implemented bufferless

deflection routing topology. Moreover, because deflection signals only propagate from inner cylinders to outer ones (decreasing c), and because buffers are not used, all Data Vortex topologies operating under any of the schemes discussed above are categorically livelock-free for all possible traffic patterns. More formally, at any timeslot t (defined canonically), the packets $\{P\}$ within the network can be partitioned into C batches N_c , each given by the packets' positions on the cylinders. Because deflection signals $d(p_i, p_j)$ can only be sent from a packet with a higher batch number to one with a lower number (i.e., $i = j - 1 \forall d(p_i, p_j)$), the evacuation time of the network can be loosely bounded by $T \leq PH_{\max} = P(2 \log_2 H + A)$, where H_{\max} represents the largest possible hop count [16], [17], [18], [19].

2.7 Traffic Parameters

For the following discussions, the probability that an input node is blocked is annotated as ρ , and the probability that a given input interface has a packet load to offer to the network is given by λ . The likelihood that a node within the network is occupied by a packet at a particular timeslot is γ and, thus, by the Law of Large Numbers [20], the total number of packets in the system at a given timeslot is approximately $P \approx \gamma A H (\log_2 H + 1)$. A probability distribution for the number of hops n experienced by packets within a Data Vortex topology of a particular size (A, C, H) under a load λ is therefore written as $\wp_{(A,C,H),\gamma}(n)$, with mean $\bar{n}_{(A,C,H),\gamma}$.

3 ANALYTICAL CHARACTERIZATION

In order to perform an algebraic analysis of the latency and throughput characteristics of the Data Vortex architecture, a few minor assumptions must first be made. First, as with nearly all analytic characterizations, in particular, mathematically well-behaved offered traffic patterns yield the simplest closed-form solutions; this analysis considers only uniformly distributed, uniformly addressed, identically and independently distributed Bernoulli traffic. Packets are said to propagate in canonical discrete time units, and nodes are treated as simple, single-packet 2×2 routing switches.

Moreover, this analysis assumes that all nodes in the topology have equal probability of being occupied by a packet, regardless of depth or position. The Law of Large Numbers [20] is heavily utilized in order to simplify the analysis and to reveal fundamental symmetries in the routing structure. It is also assumed that the traffic parameters are locally time-invariant. While some of these approximations are relatively coarse, it is later shown that the algebraically derived results are almost identical to the simulated results for the throughput and latency relationships.

3.1 Stochastics

Although it is not entirely deterministic, the routing of the Data Vortex architecture can easily be described within the framework of modern stochastics. Each routing pathway is a simple D/1/1 queue (by the nomenclature of Kendall and Lee), and each switching node has a well defined transition matrix.

Let us first define the nomenclature *north* (\mathbf{n}) and *west* (\mathbf{w}) to be the inputs of a node from an upstream cylinder and from the same cylinder, respectively. *East* (\mathbf{e}) and

south (\mathbf{s}) thus describe outputs from a node to the same cylinder and to a downstream cylinder, respectively. By the conservation of probability,

$$\begin{aligned} \Pr(\mathbf{n} : \mathbf{e}) + \Pr(\mathbf{n} : \mathbf{s}) &= 1 \\ \Pr(\mathbf{w} : \mathbf{e}) + \Pr(\mathbf{w} : \mathbf{s}) &= 1. \end{aligned}$$

Now, consider two events \mathbf{R} and \mathbf{D} for the address-match routing and deflection, respectively, of the packet incident on a particular node. First, southward ingress requires that no deflections occur,

$$(\mathbf{n} : \mathbf{s}) \vee (\mathbf{w} : \mathbf{s}) \leftarrow \neg \mathbf{D},$$

and a routing from north to south requires both that no deflections occur and that the address be appropriate,

$$(\mathbf{n} : \mathbf{s}) \leftarrow \mathbf{R} \wedge \neg \mathbf{D}.$$

In terms of probabilities,

$$\begin{aligned} \Pr(\mathbf{n} : \mathbf{s}) + \Pr(\mathbf{w} : \mathbf{s}) &= 1 - d \\ \Pr(\mathbf{n} : \mathbf{s}) &= r \cdot (1 - d), \end{aligned}$$

where $r = \Pr(\mathbf{R})$ and $d = \Pr(\mathbf{D})$. For i.i.d. Bernoulli traffic (i.e., $r = 1/2$), the matrix of transition probabilities is accordingly

$$T = \frac{1}{2} \begin{pmatrix} 1 + d & 1 - d \\ 1 + d & 1 - d \end{pmatrix}.$$

3.2 All-Angle Injection

First, consider the injection of packets into the first cylinder $c = 0$ of the Data Vortex topology. Deflection signals are sent from the nodes on this cylinder when they intend to occupy another node also on this outer cylinder. Such a deflection signal would prevent a queued packet from entering the system. The probability with which this situation occurs at a particular node $n_1 = (\text{mod}_A a_0 + 1, 0, G_c(h_0))$ is given by ρ , and it is equal to the probability that a packet at node $n_0 = (a_0, 0, h_0)$ is either routed or deflected to node n_1 . Deflections occur when a downstream packet is routed, which is about half of the time for uniform Bernoulli traffic, so a recursive expression for the probability of input blocking could be written as

$$\begin{aligned} \rho &= \gamma \cdot \left(\frac{1}{2} + \gamma \cdot \left(\frac{1}{2} + \gamma \cdot \left(\frac{1}{2} + \dots \right) \right) \right) \\ &= \frac{1}{2} \gamma \cdot \left(1 + \frac{1}{2} \gamma \cdot \left(1 + \frac{1}{2} \gamma \dots \right) \right) \approx \frac{\gamma}{2 - \gamma}. \end{aligned}$$

And, accordingly, for a large network, the probability of receiving a deflection signal on the middle cylinders ($1 \leq c \leq C - 2$) is approximately the same as receiving a deflection signal on the outer cylinder:

$$d \approx \rho \approx \frac{\gamma}{2 - \gamma}.$$

These expressions imply that for an offered load probability λ , the network's accepted packet load per unit time is

$$\theta_{in} = N_t \lambda (1 - \rho) \approx A H \lambda \left(1 - \frac{\gamma}{2 - \gamma} \right).$$

However, in order to determine the network throughput, the quantity of emerging packets must also be established. When angle-resolution routing is implemented, packets must circulate through entire rings of the innermost cylinder (i.e., $(\bullet, C-1, h_0)$) in order to find their destination addresses. Thus, the outflow of packets is limited to

$$\theta_{out} = N_t \frac{\gamma}{A} = H \gamma.$$

The value of the node occupancy factor γ can therefore shift depending upon the instantaneous injection to and ejection from the network. For continuous traffic streams, however, the steady-state occupancy factor can be derived by assuming the total number of packets within the network to be constant:

$$\begin{aligned} \dot{P} = 0 &\Rightarrow \theta_{in} = \theta_{out} = \Theta \\ AH\lambda \left(1 - \frac{\gamma}{2 - \gamma}\right) &\approx H\gamma \\ \Rightarrow \gamma &= A\lambda + 1 - \sqrt{A^2\lambda^2 + 1}. \end{aligned}$$

Accordingly, the total steady-state network throughput for uniform i.i.d. Bernoulli traffic can be expressed as

$$\Theta(\lambda) = H \left(A\lambda + 1 - \sqrt{A^2\lambda^2 + 1} \right),$$

which implies that deflections diminish the network throughput by an amount

$$\Delta = H \sqrt{A^2\lambda^2 + 1} - H.$$

Last, the steady-state probability of successful injection for offered Bernoulli traffic load is then

$$1 - \rho = \frac{A\lambda + 1 - \sqrt{A^2\lambda^2 + 1} - H}{A\lambda},$$

for a Data Vortex architecture utilizing all $N_t = AH$ input and output terminals.

3.3 Single-Angle Injection

It is also interesting to consider Data Vortex systems which use only a single angle on the outer cylinder for input injection. This configuration allows for the angular dimension to be used entirely for buffering, dramatically increasing the packet acceptance figure. In order to maintain an equal in-degree and out-degree for the whole network, the addresses of the nodes at different angles within the inner cylinder are allowed to become degenerate, so that all nodes $(\bullet, C-1, h_0)$ can be used as output ports for the same destination addresses; i.e., angle-resolution routing is removed from the architecture. The number of input and output terminals is thus reduced to $N_t = H$.

In such a system, the injection throughput parameters are vastly different from those in the all-angle injection case discussed above, although much of the mathematics are similar. First consider the maximum outflow, which is no longer limited by the angle-resolution routing circulations:

$$\theta_{out} = AH \gamma.$$

The inflow is now substantially higher as well, since input blocking requires that packets be deflected around an

entire outer tour. For a large network, a reasonable estimate of the number of accepted packets per unit time is

$$\rho(A) \approx \left(\frac{1}{2} + \gamma \cdot \left(\frac{1}{2} + \gamma \cdot \left(\frac{1}{2} + \dots \right)^{A-2} \right)^{A-1} \right)^A,$$

wherein the finite extent of the recursion should require only nonnegative exponents (empirical observations suggest that only about $A/2$ of the terms must be used in order to obtain a fairly precise numerical solution for common parameter values). This recursively transcendental relationship makes no closed-form solution possible, but numerical methods can be used in order to find individual solutions for integral angle number values. With only a single angle used for input injection,

$$\theta_{in} = H \lambda [1 - \rho(A)].$$

Again, equating the inflow and outflow expressions to find the steady-state throughput for uniform i.i.d. Bernoulli traffic,

$$\begin{aligned} \dot{P} = 0 &\Rightarrow \theta_{in} = \theta_{out} = \Theta \\ \lambda [1 - \rho(A)] &\approx A \gamma(A), \end{aligned}$$

which still requires a numerical solution because of its transcendental origins. Under these load conditions, deflections are quite sparse. Approximating the acceptance recursion for the single-angle injection case,

$$d(A) \approx \frac{\gamma(A)}{\frac{1}{2} - \gamma(A)} \approx 1 - \gamma(A),$$

which is difficult to express concisely; numerical methods can again be used to solve the appropriate nonclosed form expressions.

3.4 Latency Distribution

Next, a closed-form expression for the latency distribution of the Data Vortex topology as a function of the network size and congestion is found. This distribution can be divided into three distinct and independent probability distributions which can later be convolved to find the complete latency distribution (see Markov process theorem, Stieltjes integral, etc. [20], [21]).

Consider the latency distribution of the Banyan-style binary tree-based addressing structure. In the Data Vortex topology, each binary mismatch requires an extra node hop within the cylinder whereas each correct matching allows the packet to ingress to the next cylinder. Thus, in order to propagate from the outer cylinder ($c = 0$) to the innermost cylinder ($c = C - 1$), between $C - 1$ and $2C - 2$ hops are required for a packet in a completely unloaded system (i.e., no deflections). The hop probability distribution for this process, which defines the probability of an arbitrary packet traversing exactly n nodes (taking n node hops), is of course binomial:

$$\Pi_C(n) = \frac{1}{2^{C-1}} \binom{C-1}{n-C+1}, n \in [C-1, 2C-2].$$

The additional hops added to a packet's path due to circulation within the innermost cylinder for angular

routing follow an even simpler expression. For randomly-addressed traffic, each angle has an equal chance of matching the packet's destination address:

$$\Pi_A(n) = \frac{1}{A}, n \in [0, A - 1].$$

Last, the hop distribution is effectively skewed away from zero by congestion: some packets must additionally travel an even number of hops because of deflections. Packets encounter multiple deflection signals with exponentially decreasing likelihood, as represented by the probability distribution

$$\Pi_d(n) = (1 - \gamma) \sum_{i=0}^{\infty} d^i \delta(n - 2i), n \in [0, \infty],$$

where δ represents the impulse delta function, and d depends upon the injection scheme used ($q.v.$).

The component distribution functions described above are entirely independent of each other, so they can be convolved to find the total packet latency probability distribution:

$$\wp_{(A,C,H),\gamma}(n) = \Pi_C(n) \otimes \Pi_A(n) \otimes \Pi_d(n), n \in [0, \infty].$$

Moreover, the component distributions' independence allows for the mean of this distribution to be easily found:

$$\bar{n}_{(A,C,H),\gamma} = \langle n\Pi_C(n) \rangle + \langle n\Pi_A(n) \rangle + \langle n\Pi_d(n) \rangle.$$

Specifically, for a network without angle-resolution routing and with maximal input injection ($\lambda = 1$), the average latency can be expressed as

$$\bar{n}_{(A,C,H),\gamma,a'=0} = \frac{3C + 1}{2} + \frac{2d}{1 - d} = \frac{3 \log_2 H + 4}{2} + \frac{2d}{1 - d},$$

and with angle-resolution routing and the utilization of all input terminals, the average latency becomes

$$\bar{n}_{(A,C,H),\gamma,a} = \frac{3 \log_2 H + A + 3}{2} + \frac{2d}{1 - d},$$

both of which are fundamentally on the order of the logarithm of the number of input ports (i.e., $\bar{n}_{(A,C,H),\gamma} \propto O(\log_2 H)$, $A \ll H$), similar to conventional Banyan networks, with a small addition due to routing deflections and possible angle-resolution routing.

4 NETWORK PERFORMANCE SIMULATIONS

Given the modeling assumptions presented in the earlier parts of this work, designing a network simulator for the Data Vortex is relatively straight-forward and is necessary to view effects like impact on network performance by angle size, network performance under synthetic and trace-based traffic loads, and same-sized network performance comparisons with other networks.

4.1 Data Vortex Network Simulation

First, in order to determine the effect of the angle value on the Data Vortex network performance and to help select an optimal value for the angle number, a series of simulations are run. In all simulations involving the Data Vortex, it is assumed that all packets are exactly one cycle in length (i.e.,

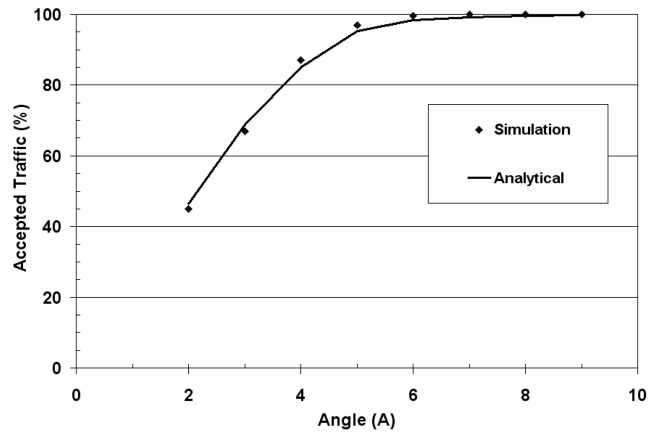


Fig. 5. Accepted traffic versus angle size for maximum random workload. The network simulated has $H = 2,048$ inputs and maximum load. For angle sizes greater than 6, the network accepts more than 99.99 percent of all traffic offered, even under maximum load.

they are only in one node at the start of any given cycle), each message is composed of exactly one packet, and packets have a randomly-chosen (or bit-reversed, for bit-reversal traffic) destination address. Likewise, it is assumed that each link has the same physical latency (one hop) and latency is computed as the time of flight in hops along the identical fiber links between optical switches (i.e., switching time is negligible compared to time of flight along the length of fiber). In addition, packets are only injected on one angle. Thus, the inputs to the Data Vortex are along the height of angle zero, and the other $A-1$ angles are used as virtual buffers. The traffic patterns used are synthetically generated as a randomly-chosen input address and either a randomly-chosen output address (for random traffic workloads) or a bit-reversed output address (for bit-reversal workloads). The bit-reversed output address is found by simply reversing the order of the input address bits ($h_n h_{n-1} \dots h_0 \rightarrow h_0 \dots h_{n-1} h_n$).

When injecting on one angle only, the different-sized Data Vortex networks exhibit very similar plots for accepted traffic ratio versus a scaled workload, so a network size (height) of 2,048 is selected for illustration. As can be seen by the results in Fig. 5, the angle value affects the successful packet injection ratio as well as the average packet latency (as shown in Fig. 6). The network is simulated while under a maximum load, meaning that an attempted packet injection occurs at each node along the height of angle 0 on every cycle. These results closely correlate with the projected results from the stochastic analysis above, as shown in the plots.

As the plots illustrate, changing the angle value from 2 to 6 while keeping all other network parameters constant increases packet acceptance by about 100 percent and decreases latency by about 40 percent. This shows the serious effect that an undersized angle value has on network performance. Based on the experimental data, an angle size of 5 to 6 is optimal for injection on one angle, given the trade-off between entire network switch fabric size/cost and acceptance. It should be noted that the resultant angle parameter of 6 or greater is to attain a packet rejection rate of 0.01 percent or lower under maximum load, and it is only valid for network setups that inject upon one

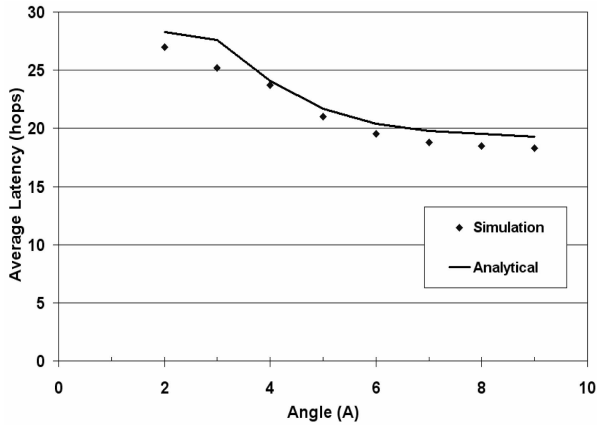


Fig. 6. Average latency versus angle size for random workload. The network simulated has $H = 2,048$ inputs. The average latency drops with increased angle size, with a reasonably low latency value corresponding to $A = 6$.

angle. The plots also illustrate the very close correlation of simulation results with the stochastic analysis earlier.

4.2 Butterfly and Omega Network Simulations

Multistage packet-switched interconnection networks such as the butterfly and omega networks are usually input and output blocking networks (unlike the Data Vortex, which only exhibits input blocking). Deflection routing can be used in these networks, but the performance of deflection routing with no buffering in networks not designed specifically for this method of routing is greatly reduced [34]. Both the omega and butterfly networks are simple and were designed for electrical store-and-forward routing. Each could be easily adapted to optical store-and-forward packet switching if efficient random-access all-optical memory without the need for O/E and E/O conversion could be used. However, output and intra-network blocking in these networks with store-and-forward routing present the need for data to be buffered for a number of cycles. Butterfly and omega networks therefore include the need to perform O/E and E/O conversions in order to buffer data electronically, as efficient all-optical buffering is not currently available [22]. The two comparison networks (omega and butterfly) are chosen to represent simple and widely-researched examples of known indirect multistage supercomputer interconnection networks. The Data Vortex can be compared to an adaptation of a more similar (i.e., input-blocking and previously-applied to photonic switching) network such as the shufflenet [33], but the comparison would involve many more variables and be less clear to the reader. Both networks (optical implementations of the omega and butterfly) are compared to the Data Vortex in performance simulations later in the document. In order to compare optical implementations of butterfly and omega networks to the Data Vortex, certain assumptions have to be made.

First, it is assumed that the buffering necessary for an all-optical butterfly or omega implementation is efficient and fast enough to ignore the buffering time and the inherent decrease in switching speed that accompanies O/E and E/O conversion. This is not entirely a valid assumption, as buffering does increase switching time, increase switch complexity, and even consumes more power. However, this yields a straight comparison of the number of hops (i.e., the time of flight) of

packets throughout the respective networks and neglects switching time, under the assumption that efficient optical buffers will be implemented in the near future for blocking topologies such as butterfly and omega networks. It should be noted that under current technological constraints, however, a comparison hop in the Data Vortex is shorter in time than those of the butterfly and omega networks (a point to be kept in mind when viewing the simulation data for latency). This is because the hop as compared in the three networks consists simply of the same-length fiber link with a fixed delay as the “hop” unit and ignores the O/E/O conversion in the store-and-forward comparison networks. This is done for a straighter comparison that is more independent of technology and network size, as the O/E/O conversion time depends not only on the technology used, but also on the packet size and number of WDM channels used.

In addition, the same assumptions from the Data Vortex simulations apply to the butterfly and omega simulations—all packets are exactly one cycle in length (i.e., they are only in one node at the start of any given cycle), each message is composed of exactly one packet, and packets have a randomly-chosen or bit-reversed destination address. With these assumptions made, the results for comparison simulations are shown in the next section.

Finally, assumptions about the structure of the omega and butterfly networks are made as well. Each network is assumed to buffer one data packet at each output of its constituent 2×2 crossbar switches at each stage of the network. If another packet is in contention for an output that is currently buffering a packet, the newcomer is blocked and remains buffered in its original node (exhibiting output blocking). This is a fair assumption, as most current implementations of each network buffer at least one packet at each output, and more than one packet buffered would be an egregious violation of the previous assumption that buffering and switching times are negligible. Once all assumptions are made, a relatively fair comparison of the three networks can be made, as shown in the next section.

4.3 Latency Comparison

The average latency is computed in each network the same way—as the number of hops or links the packets must traverse from input to output. The latency measurements only include latency of packets within the network, and packets blocked before reaching the first stage (the input) of the network are assumed to be dropped and potentially injected again later. The latency measurements for the two output blocking architectures count cycles that data packets are buffered as hops as well, as buffered data waits one cycle before attempting again to ingress to the next stage. The three networks exhibit average latency values as shown in Fig. 7.

As indicated by the plots, the Data Vortex exhibits similar latency values on average to those of the butterfly and omega networks for random traffic loads and much lower latency values on average for bit-reversal traffic loads. As mentioned previously, the latency of each hop on the Data Vortex architecture could be substantially lower than the latency presented in a hop in either of the comparison networks, however, as switching in the photonic Data Vortex does not involve the time required by O/E and E/O conversions necessary in the butterfly and omega networks, which were ignored. Thus, the Data Vortex has average packet latency that is comparable to each of the other networks.

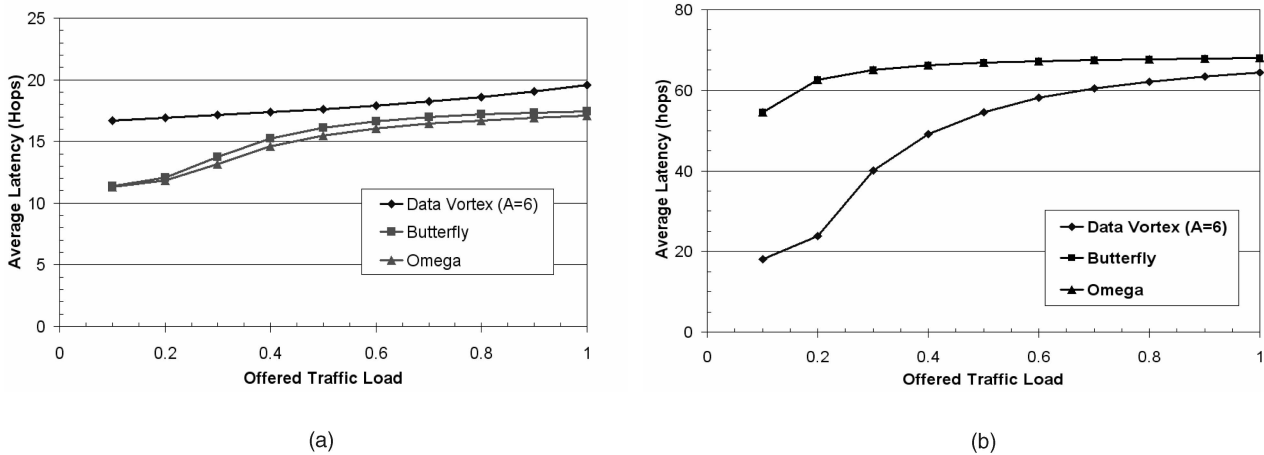


Fig. 7. Average latency versus offered traffic load for 2,048 inputs. The average latency of the data vortex for random traffic (a) is only slightly higher, and it should be noted that “hops” within the data vortex are actually shorter than in the other two networks due to simpler switching and no O/E and E/O conversions for buffering. (b) Shows that the data vortex exhibits a much lower latency for bit-reversal traffic than the two comparison networks, which both exhibit very similar (overlapping) latency curves, due to their similar structures and address resolution schemes.

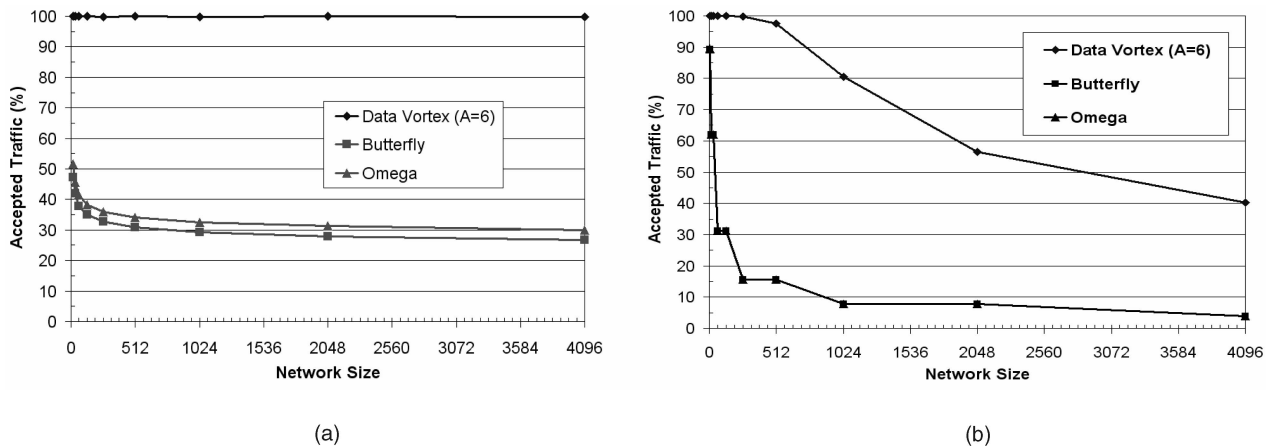


Fig. 8. Accepted traffic versus network input size for 40 percent load. For random traffic loads (a), the acceptance of the data vortex is more than 20 percent higher in packet acceptance for small networks and maintains close to 100 percent acceptance, in contrast to the decline in packet acceptance by the two comparison networks as network size increases. For bit-reversal traffic workloads (b), the data vortex accepts more packets even for small network sizes and over eight times as many packets as the two comparison networks for larger networks.

4.4 Injection Ratio Comparison

The injection ratio for each network is measured as the ratio of successful injections to attempted injections. As mentioned previously, for comparison to the other networks the Data Vortex is only injected upon on one angle, making the height value equal to the number of inputs to the network. Thus, the same number of packet injections is attempted in each of the networks for a given input size and network load. The simulation results are as shown in Fig. 8 for varying network input sizes and 40 percent load, and the results are shown in Fig. 9 for a fixed size of 2,048 inputs and varying load. The load of 40 percent was selected because the two comparison networks saturate at about 50 percent load, whereas the Data Vortex does not, so any comparison above 40 percent would be unfair. As the plots illustrate, the Data Vortex accepts about twice as many packets as the comparison networks when offered the same workload. This higher acceptance rate is due partially to the fact that the Data Vortex has fewer potential data packet collisions within the network due to its always-moving nature and nonblocking switches. Even when under maximum load and deflections within the network are more common, the Data Vortex utilizes the virtual buffering

provided by the additional angles to accommodate more data packets while maintaining latencies comparable to the other networks. Likewise, due to the lack of need for O/E and E/O signal conversions, the switching is faster, simpler, and more power efficient. This makes the addition of angles fair, as the three networks thus have similar costs.

5 CONCLUSION

In order for large-scale parallel computers to be designed and constructed, an interconnection network that can handle a large workload with minimum latency is required. The Data Vortex was shown to have similar average latency in hops per data packet to two widely-accepted existing network architectures. As conversion time delays from optical to electrical and back for buffering were not included in the measure of “hops” and buffering time itself was also ignored for the two comparison networks, the Data Vortex is potentially even lower in latency than as compared to the two comparison networks. The Data Vortex, by design, exhibits neither type of delay, and the delays in the other two networks are switch hardware dependent, making them hard to accurately include in time

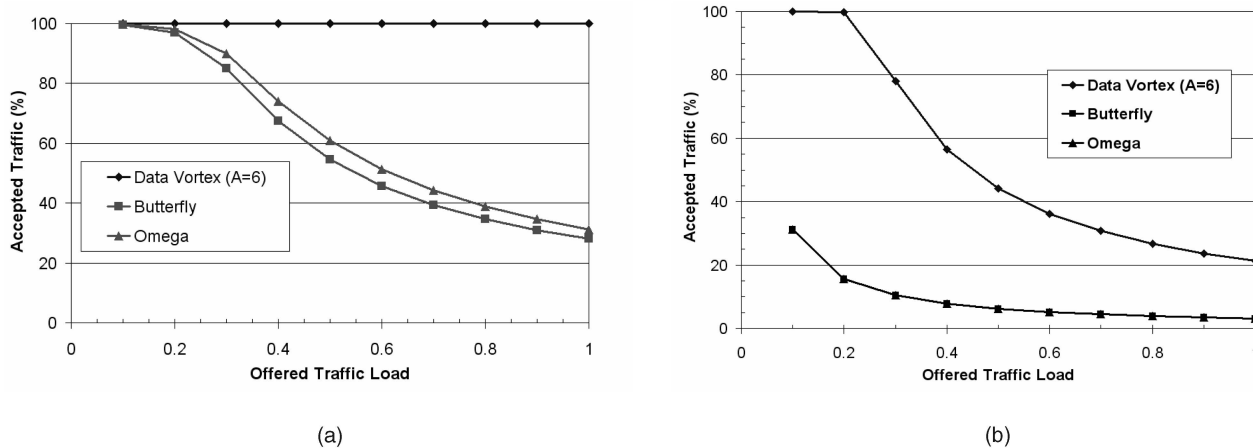


Fig. 9. Accepted traffic versus offered traffic for a fixed input/output size of 2,048. The acceptance of the data vortex remains much greater than those of the comparison networks, maintaining nearly 100 percent acceptance for random traffic workloads (a) and still accepts over three times as much traffic for bit-reversal traffic workloads (b).

estimation for comparison. The Data Vortex was also shown to have roughly twice as much packet acceptance for the same given 50 percent load workload and network size and three times as much packet acceptance for the same 100 percent load workload and network size. Therefore, the Data Vortex greatly outperforms the comparison networks in simulations using the metrics of latency and packet acceptance. Additionally, the angle value of the Data Vortex was studied and found to have a tremendous impact on network performance.

Interconnection network latency has a major impact on the performance and usability of a supercomputer. Advances in processors and memory offer the opportunity for tremendous performance. But, without comparable improvements in interconnection networks, this potential will be lost for most applications. Topologies like the Data Vortex offer the opportunity to harness the minimum latency and enormous bandwidth potential of DWDM optical technology. Out of these new topologies, a new generation of supercomputer will be born.

ACKNOWLEDGMENTS

The authors would like to thank John P. Mack for generating the three-dimensional illustrations of the Data Vortex topology and the reviewers of this paper for their thoughtful comments. This work was supported in part by the Department of Defense under contract MDA-904-03-C-0471.

REFERENCES

- [1] M. Yokokawa, "Present Status of Development of the Earth Simulator," *Innovative Architecture for Future Generation High-Performance Processors and Systems*, pp. 93-99, Jan. 2001.
- [2] X. Shen, F. Yan, and Y. Pan, "Equivalent Permutation Capabilities Between Time-Division Optical Omega Networks and Non-Optical Extra-Stage Omega Networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 518-524, Aug. 2001.
- [3] C. Reed, "Multiple Level Minimum Logic Network," US Patent 5 996 020, Nov. 1999.
- [4] Q. Yang, K. Bergman, G.D. Hughes, and F.G. Johnson, "WDM Packet Routing for High-Capacity Data Networks," *J. Lightwave Technology*, vol. 19, no. 10, pp. 1420-1426, Oct. 2001.
- [5] G.I. Papadimitriou, C. Papazoglou, and A.S. Pomportsis, "Optical Switching: Switch Fabrics, Techniques, and Architectures," *J. Lightwave Technology*, vol. 21, no. 2, pp. 384-405, Feb. 2003.
- [6] Q. Yang and K. Bergman, "Performances of the Data Vortex Switch Architecture under Nonuniform and Bursty Traffic," *J. Lightwave Technology*, vol. 20, no. 8, pp. 1242-1247, Aug. 2002.
- [7] Q. Yang and K. Bergman, "Traffic Control and WDM Routing in the Data Vortex Packet Switch," *IEEE Photonics Technologies Letters*, vol. 14, no. 2, pp. 236-238, Feb. 2002.
- [8] W. Lu, K. Bergman, and Q. Yang, "WDM Routing with Low Cross-Talk in the Data Vortex Packet Switching Fabric," *Proc. Optical Fiber Conf. (OFC '03)*, vol. 2, FS4, pp. 795-797, Mar. 2003.
- [9] W. Lu, B.A. Small, O. Liboiron-Ladouceur, J.N. Kutz, and K. Bergman, "Optical Packet Switching through Multiple Nodes in the Data Vortex Architecture," *Proc. Ann. Meeting IEEE Lasers and Electro-Optics Soc. (LEOS '03)*, vol. 1, MF2, pp. 53-54, Oct. 2003.
- [10] W. Lu, B.A. Small, K. Bergman, and L. Leng, "Ultra-High Capacity WDM Optical Packet Routing through an 8-Node Data Vortex Sub-Network," *Proc. Optical Fiber Conf. (OFC '04)*, MF94, pp. 281-283, Mar. 2004.
- [11] W. Lu, B.A. Small, J.P. Mack, L. Leng, and K. Bergman, "Optical Packet Routing and Virtual Buffering in an Eight-Node Data Vortex Switching Fabric," *IEEE Photonics Technology Letters*, vol. 16, no. 8, pp. 1981-1983, Aug. 2004.
- [12] G.P. Agrawal, *Fiber-Optic Communication Systems*, third ed. Wiley & Sons, 2002.
- [13] N. Biggs, *Algebraic Graph Theory*, second ed. Cambridge Univ. Press, 1993.
- [14] A. Shacham, B.A. Small, O. Liboiron-Ladouceur, J.P. Mack, and K. Bergman, "An Ultra-Low Latency Routing Node for Optical Packet Interconnection Networks," *Proc. 17th Ann. LEOS Meeting*, paper WM2, pp. 565-566, Nov. 2004.
- [15] Q. Yang, "Optical Packet Switching for High Performance Computing," PhD dissertation, Princeton Univ., Jan. 2002.
- [16] W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [17] J.T. Brassil and R.L. Cruz, "Bounds on Maximum Delay in Networks with Deflection Routing," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 7, pp. 724-732, July 1995.
- [18] B. Hajek, "Bounds on Evacuation Time for Deflection Routing," *Distributed Computing*, vol. 5, no. 1, pp. 1-6, 1991.
- [19] A.K. Choudhury and V.O.K. Li, "Effect of Contention Resolution Rules on the Performance of Deflection Routing," *Proc. GlobeCom '91*, vol. 3, pp. 1706-11, Dec. 1991.
- [20] W. Feller, *An Introduction to Probability Theory and Its Applications*, third ed. Wiley, 1968.
- [21] I.I. Hirschman and D.V. Widder, *The Convolution Transform*. Princeton Univ. Press, 1955.
- [22] X. Liu and Q.-P. Gu, "Multicasts on WDM All-Optical Butterfly Networks," *J. Information Science and Eng.*, vol. 18, no. 6, pp. 1049-1058, Nov. 2002.
- [23] B.A. Small, A. Shacham, K. Bergman, K. Athikulwongse, C. Hawkins, and D.S. Wills, "Emulation of Realistic Network Traffic Patterns on an Eight-Node Data Vortex Interconnection Network Subsystem," *J. Optical Networking*, vol. 3, pp. 802-809, Nov. 2004.

- [24] W. Lu, B.A. Small, J. Mack, K. Bergman, and L. Leng, "Ultra-High Capacity WDM Optical Packet Routing through an 8-Node Data Vortex Sub-Network," *Proc. Optical Fiber Conf. (OFC '04)*, poster MF94, pp. 281-283, Mar. 2004.
- [25] B.A. Small, O. Liboiron-Ladouceur, A. Shacham, J.P. Mack, and K. Bergman, "Demonstration of a Complete 12-Port Terabit Capacity Optical Packet Switching Fabric," *Proc. Optical Fiber Conf. (OFC '05)*, paper OWK1, Mar. 2005.
- [26] A. Shacham, B.A. Small, O. Liboiron-Ladouceur, and K. Bergman, "A Fully Implemented 12×12 Data Vortex Optical Packet Switching Interconnection Network," *J. Lightwave Technology*, vol. 23, no. 10, pp. 3066-3075, Oct. 2005.
- [27] B.A. Small, A. Shacham, and K. Bergman, "Ultra-Low Latency Optical Packet Switching Node," *IEEE Photonics Technology Letters*, vol. 17, no. 7, pp. 1564-1566, July 2005.
- [28] B.A. Small and K. Bergman, "Slot Timing Considerations in Optical Packet Switching Networks," *IEEE Photonics Technology Letters*, vol. 17, no. 11, pp. 2478-2480, Nov. 2005.
- [29] TOP500.org, "TOP500 List for June 2005," <http://www.top500.org/lists/current.php>, June 2005.
- [30] M. Yokokawa, "Present Status of Development of the Earth Simulator," *Innovative Architecture for Future Generation High-Performance Processors and Systems*, pp. 93-99, Jan. 2001.
- [31] N.F. Maxemchuk, "Comparison of Deflection and Store-and-Forward Techniques in the Manhattan Street and Shuffle-Exchange Networks," *Proc. Eighth Ann. Joint Conf. IEEE Computer and Comm. Soc. (INFOCOM '89)*, vol. 3, pp. 800-809, Apr. 1989.
- [32] G. Albertengo, R. Lo Cigno, and G. Panizzardi, "The Deflection Network: A Reliable High Speed Packet Network for Computer Communication," *Proc. Fifth Ann. European Computer Conf. (CompEuro '91)*, pp. 84-88, May 13-16, 1991.
- [33] M.G. Hluchyj and M.J. Karol, "ShuffleNet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks," *Proc. IEEE Seventh Ann. Joint Conf. IEEE Computer and Comm. Soc. (INFOCOM '88)*, pp. 379-390, Mar. 1988.
- [34] A.S. Acampora and S.I.A. Shah, "Multihop Lightwave Networks: A Comparison of Store-and Forward and Hot-Potato Routing," *IEEE Trans. Comm.*, vol. 40, no. 6, pp. 1082-1090, June 1992.



Cory Hawkins (S'99) received the BS degree in computer engineering with highest honor in 2000, the MS degree in electrical and computer engineering in 2001, and is currently pursuing the PhD degree in electrical and computer engineering, all at the Georgia Institute of Technology in Atlanta, Georgia. His current research interests are in the areas of interconnection network architectures for parallel computers, routing algorithms, and embedded micronetworks for system-on-a-chip applications. He is a student member of the IEEE, a member of the IEEE Computer Society, and a member of the IEEE Communications Society.



Benjamin A. Small (S'98, M'06) received the BS (with honor) and MS degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2001 and 2002, respectively. He received the MPhil and PhD (with distinction) degrees from Columbia University in the City of New York, both in 2005. He is currently a postdoctoral researcher at Columbia. His interests include optoelectronic and photonic device physics and modeling, specifically relating to high-bandwidth optical communications in the context of low-latency interconnection networks. He also has done work in the area of multiple-stage interconnection network behavior and traffic analysis, particularly for architectures based on photonic and fiber optic technologies. Dr. Small is a member of the IEEE and a member of the IEEE Lasers and Electro-Optics Society.



D. Scott Wills (S'79-M'90-SM'98) received the BS degree in physics from the Georgia Institute of Technology in 1983 and the SM, EE, and ScD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in 1985, 1987, and 1990, respectively. He is an associate professor of electrical and computer engineering at the Georgia Institute of Technology. His research interests include short wire VLSI architectures, high-throughput portable processing systems, architectural modeling for gigascale (GSI) technology, and high-efficiency image processors. Dr. Wills is a senior member of the IEEE, a member of the IEEE Computer Society, and an associate editor for the *IEEE Transactions on Computers*.



Keren Bergman (S'87-M'93) received the BS degree in electrical engineering from Bucknell University, Lewisburg, Pennsylvania, in 1989 and the MS and PhD degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1991 and 1994, respectively. She is a professor of electrical engineering at Columbia University. Her research interests include ultrafast optical signal processing, WDM optical networking, and optical packet interconnection for high-performance computing. Dr. Bergman is a member of the IEEE and a fellow of the Optical Society of America (OSA). She currently serves as associate editor for *IEEE Photonic Technology Letters* and for the *OSA Journal of Optical Networking*.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.