Traffic Control and WDM Routing in the Data Vortex Packet Switch

Qimin Yang, *Student Member, IEEE,* and Keren Bergman, *Member, IEEE*

***Abstract*—A traffic control mechanism that eliminates the need for optical buffering and simplifies the routing procedure is experimentally demonstrated within the Data Vortex packet switch. The hierarchical switching system also employs a wavelength-division-multiplexing header-encoding technique to minimize the header processing and routing latency at each hop.**

***Index Terms*—Photonic packet switching, wavelength encoding.**

I. Introduction

PHOTONIC packet switching has emerged as an attractive way to overcome the electronic bottleneck in today's fast growing communication and computing systems [1]. As more bandwidth is provided in these systems, issues such as processing, buffering, scalability, and latency remain the key design challenges for the switching network elements. A new architecture called the Data Vortex was, thus, proposed that is uniquely free of optical buffers and enables simple routing logic for large scale low latency packet switch fabrics [2]. The hierarchical system employs a synchronous timing and distributed control signaling to avoid packet contention and to achieve simplicity, scalability, and high throughput. In addition, it facilitates the use of a wavelength-header-encoding technique to minimize the routing function and switching latency. In this letter, we demonstrate experimentally the control mechanism between two fully functional routing nodes of the Data Vortex. In addition, the techniques of wavelength-division-multiplexing (WDM)-encoded header bits and wavelength-filtering recovery are also shown within the test bed.

A schematic of the Data Vortex architecture is shown in Fig. 1. The size of the switch fabric is defined by the parameters A and H, corresponding to the number of routing nodes lying along the "angle" and "height" dimension, respectively. The number of cylinders, C scales as $\mathrm{C} = \log_2(\mathrm{H}) + 1$. Packets are processed synchronously within the switch fabric in a parallel manner. Within each time slot, every packet progresses by one angle forward either along the solid line toward the same cylinder or along the dashed line toward the inner cylinder.

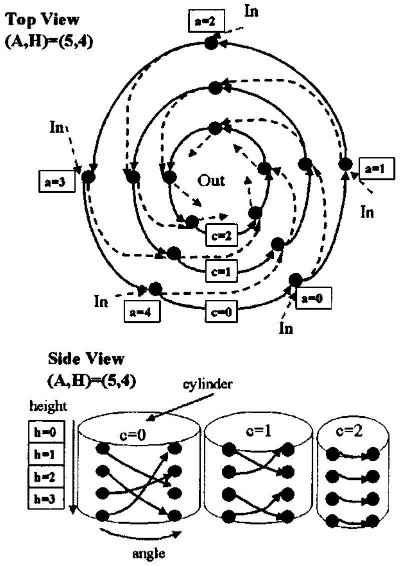


Fig. 1. Schematic of the Data Vortex topology with $\mathrm{A} = 5, \mathrm{H} = 4, \mathrm{C} = 3$. Routing tours are seen from the top and the side.

The solid routing pattern at the specific cylinder shown can be constructed as follows: divide the total number of nodes along the height, H into 2^c subgroups, where c is the index of the cylinders. The first subgroup is then mapped as follows: the top half of the remaining nodes at angle (a) are connected to the bottom half of the remaining nodes at angle $(\mathrm{a} + 1)$. This step is repeated until all nodes of the first subgroup are mapped from angle (a) to angle $(\mathrm{a} + 1)$. If multiple subgroups exist, the rest of them copy the mapping pattern of the first subgroup. The solid routing paths are repeated from angle to angle, which provide permutations between "1" and "0" for the specific header bit. The dashed-line paths between neighboring cylinders maintain the same height index h as they are only used to forward the packets. A more detailed discussion of the Data Vortex architecture is provided in [2] and [4].

Packets are injected into the outermost cylinder $(\mathrm{c} = 0)$ from the input ports, and emerge from the innermost cylinder $(\mathrm{c} = \mathrm{C} - 1)$ toward the output ports. These packets are self-routed by proceeding along the angle dimension from the outer cylinder toward the inner cylinder. Each cylinder progress will fix a specific bit of binary target header. This hierarchical routing procedure allows us to employ a technique of WDM-header encoding, by which the single-header-bit-based routing is accomplished by wavelength filtering at the header retrieval process. Since the header bits run at the rather low packet rate, there is no requirement of high-speed electronics within the node. Compared with

Manuscript received April 24, 2001; revised September 18, 2001. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) and by the National Security Agency (NSA) through an agreement with the National Aeronautics and Space Administration (NASA) under Contract 960199, and by the National Science Foundation (NSF) under Contract ECS98-00401.

Q. Yang is with the Electrical Engineering Department, Princeton University, Princeton, NJ 08540 USA (e-mail: qyang@ee.princeton.edu).

K. Bergman was with the Electrical Engineering Department, Princeton University, Princeton, NJ 08540 USA, he is now with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

Publisher Item Identifier S 1041-1135(02)00248-3.

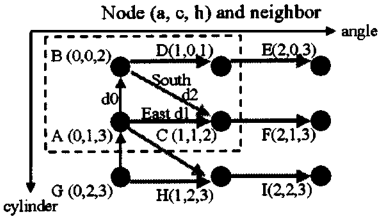


Fig. 2. Control signaling (vertical lines) between competing nodes within the neighborhood.

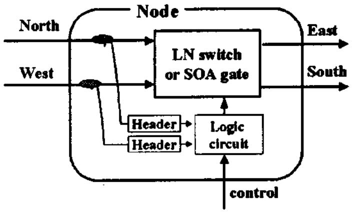


Fig. 3. Routing node structure.

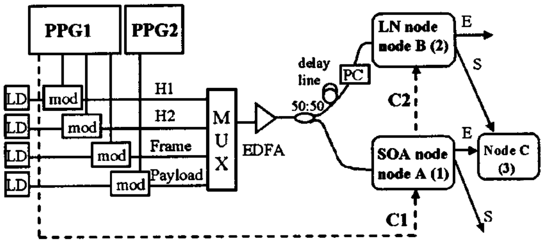


Fig. 4. Diagram of two-node routing experiment for control mechanism demonstration.

TDM-encoded header bits, the synchronization complexity can also be greatly relaxed within the packet-routing process.

A key design of the system is a distributed control signaling mechanism among routing nodes to achieve bufferless operation and simple routing logic. With the embedded synchronous timing, this scheme can schedule the traffic flow of the neighboring nodes properly so that packet conflict is eliminated. To implement the scheme, control lines are applied between any pair of nodes, which have competitive output paths to the same node.

To see this more clearly, a small group of nodes around node C (1,1,2) are shown in Fig. 2 with $\mathrm{H} = 4$ switch fabric, where each node is labeled by coordinate (a, c, and h). A specific example of control signaling (vertical line) between node A (0,1,3) and node B (0,0,2) is shown because they both send packets to node C (1,1,2). The mechanism is very simple; a "deflection" control message is automatically triggered from node A to node B whenever A sends a packet to C. Since it took a latency of d0 to deliver the control message, packet at node A must be slightly earlier than the packet at node B for proper scheduling. If both B and A have packets addressed to C, the control message is then able to prevent B's packet from progressing inside at the same packet slot. The deflected packet remains at its current cylinder instead by propagating to node D in Fig. 2. A virtual buffering mechanism is, thus, provided for the deflected packet, and only a slight latency penalty will be introduced. This is because the deflected packet recovers its direction vector toward the target every other clock cycle (in two hops) by staying on the same cylinder. The system can be kept synchronous by properly designing the link latency. As shown in Fig. 2, d1 and d2 represent the propagation latencies for the same-cylinder link (East out path) and the neighbor-cylinder link (South out path), respectively. In practical implementations, the control latency d0 only takes a small percentage of the packet period, because physically the competing nodes are located close to each other. Therefore, by simply making $\mathrm{d2} + \mathrm{d0} = \mathrm{d1}$, the switch system is able to maintain synchronous operation, as well as allow the correct setup of the control mechanism.

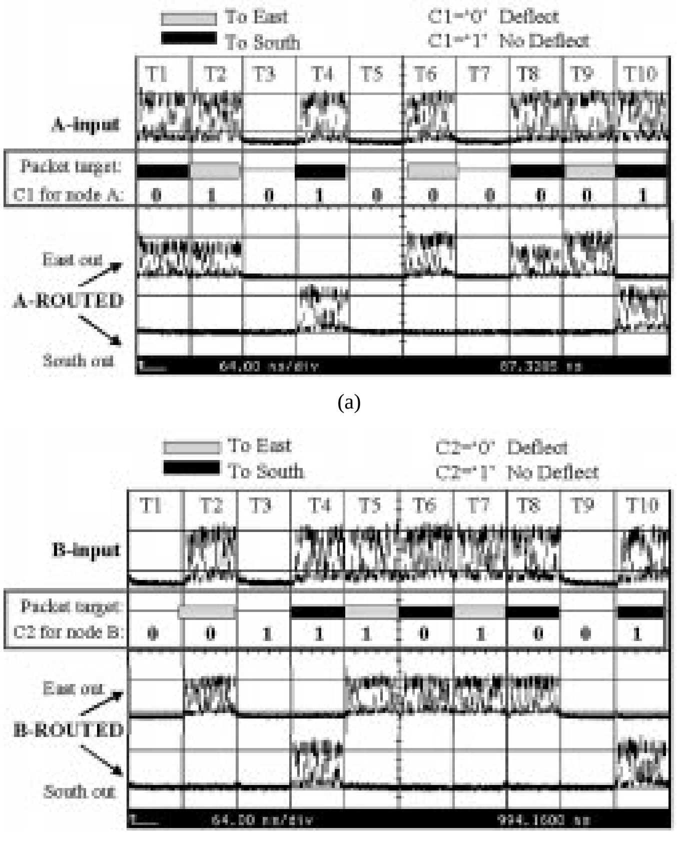
II. Experiment and Results

The routing nodes in the Data Vortex each consists of two input ports, one from an outer cylinder (North) and one from the same cylinder (West), and two output ports, one to the same cylinder (East), and one to an inner cylinder (South). As shown in Fig. 3, within each node a small amount of optical power will be tapped off for header retrieval and routing decision, which controls the state of an electrooptic switch for proper packet routing.

The basic routing details have been demonstrated previously in a recirculative testbed, using a $\mathrm{LiNbO_3}$ switch for the node (labeled node 2 in Fig. 4), that emulates routings under cascaded node hops [3], [4]. In this experiment, another node (labeled node 1 in Fig. 4) is built based upon semiconductor optical amplifiers (SOA). The decision circuitry of node 1 and node 2 are set properly to configure themselves as node A and node B in Fig. 2, respectively.

Fig. 4 shows the experimental setup. During the packet construction, two header channels at $\lambda 1 = 1550.9$ nm and $\lambda 2 = 1547.7$ nm are programmed to represent the header bits H_1 and H_2 so that $(\mathrm{H}_1\mathrm{H}_2)$ represents the binary destination address. Even though all the header fields are carried with the payload information throughout the packet switch, at each specific cylinder node, only the corresponding header bit or wavelength field needs to be decoded for routing. Node A only decodes H2 bit of the incoming packet using a 1-nm bandpass filter centered at $\lambda 2$. Similarly, node B employs a filter at $\lambda 1$ to decode the H1 bit for the routing procedure. A framing channel at $\lambda 3 = 1555.6$ nm is programmed to indicate the presence of a packet in the time slot.

The packet sequence into node A is delayed relative to the packet sequence into node B by four-packet lengths to simulate an independent packet sequence. An additional polarization controller (PC) is needed for node B due to the polarization sensitivity of the $\mathrm{LiNbO_3}$ switch. The traffic control signal (C1) for node A is programmed directly from PPG1 to allow flexible initial settings, while the control for node B (C2) is generated from the routing logic circuitry of node A. As mentioned above, a packet arriving at node B must be slightly later than the packet arriving at node A to allow the control C2 to set up. This time difference is limited by the decision circuitry delay of node A, which is measured to be 1.6 ns for the setup. With additional connection latency, C2 takes about 2–3 ns to setup, which is a small ratio of the packet period.



(a)

(b)

Fig. 5. (a) Routing result at node A. (b) Routing result at node B.

To study the control signaling mechanism under different traffic settings, we programmed a short sequence of ten packets, each carrying different routing destinations. The packets are 64 ns long, each carrying a payload modulated at 10 Gb/s on a separate wavelength $\lambda4 = 1553.3$ nm. A 6.4-ns guard time is programmed within the payload at the edge of the packet boundary allowing for the routing transients.

For a specific input sequence A-input into node A, shown at the top trace in Fig. 5(a), we randomly set its control sequence C1 to be "01010,00001," where $\mathrm{C1} = 0$ means "deflect" and C1 $= 1$ means "do not deflect" the message intended for the packet slot. Combining the destinations of each incoming packet, the output A-ROUTED can be determined by the routing logic with control signaling scheme. The routed packets at the East out and the South out are shown as the lower two traces in Fig. 5(a). As evident from these traces, the packets were routed successfully from node A in accordance with the programmed routing logic.

According to the control mechanism, a "deflect" control message is triggered whenever node A sends a packet to its East out (node C), therefore, the C2 sequence is expected to be "00111,01001." Similarly, the routing output B-ROUTED is determined by the routing destination of B-input, as well as the control signal C2. The routing result at node B is shown in Fig. 5(b), which confirms the correct implementation of the control scheme within the testbed.

We can follow for example the programmed input sequence at node B (B-input). The packets carry a South routing destination (toward node C) within the T4, T6, T8, and T10 time slots. However, this connection can be logically established only during T4 and T10 as permitted by the control mechanism, as the lowest trace shown in Fig. 5(b). This is because node A, which has priority, routed packets to node C during slots T6 and T8, triggering "deflection" for the C2 messages. The deflected packets at node B exited through the East output port.

Since every routing node has a corresponding traffic control signal, this distributed mechanism effectively eliminates the packet conflict within the whole switch fabric and greatly facilitates the optical implementation of the system.

III. Conclusion

We have demonstrated a successful testbed of control mechanism between two given routing nodes of the Data Vortex switch. The control signaling effectively eliminates the packet conflict within the architecture, and therefore, greatly simplifies the optical implementation of the system. The hierarchy structure also enables single-wavelength-based routing at each node acheiving low latency.

References

[1] R. S. Tucker *et al.*, "Photonic packet switching: An overview," *IEICE Trans. Electron.*, vol. E82-C, pp. 202–212, Feb. 1999.

[2] C. Reed, "Multiple level minimum logic network," U.S. Patent 5 996 020, Nov. 30, 1999.

[3] Q. Yang, M. F. Arend, G. D. Hughes, F. G. Johnson, and K. Bergman, "WDM/TDM optical packet switched network for supercomputing," in *Optics in Computing 2000, SPIE*, vol. 4089, pp. 1555–1561.

[4] Q. Yang, K. Bergman, F. G. Johnson, and G. D. Hughes, "WDM packet routing for high capacity data networks," *J. Lightwave Technol.*, vol. 19, pp. 1420–1426, Oct. 2001.