Performance Evaluation for Different Arrangements of Routing and Forwarding Paths within Bufferless Data Vortex Networks

Qimin Yang

Harvey Mudd College, Engineering Department, Claremont, CA 91711,
qimin_yang@hmc.edu

Abstract

Extended performance evaluation is carried out on Data Vortex optical interconnection networks with different routing and forwarding path arrangements [1]. A modified Data Vortex network architecture based on general k-ary decoding routing at each node has been proposed and different cases are compared in search for the optimum layout. For bufferless implementation, the original Data Vortex networks based on binary decoding stages are shown to achieve the best combined routing performance in throughput and latency. We specifically focus on the performance comparison between the binary decoding (k=2) and 4-ary decoding (k=4) cases to illustrate the different network behaviors. The results provide insight to how the different routing and forwarding path arrangements affect the overall network performance in throughput and latency. The binary Data Vortex networks outperform 4-ary networks even though a much smaller number of cylinder levels are required in a 4-ary network. There is only slight reduction in the average packet latency within the 4-ary network, while its deflection induced traffic backpressure under bufferless operation could greatly limit the throughput and make it less desirable. Future work may include such performance evaluation when extra buffering is available at routing nodes.

Keywords: Packet Switch, Interconnection Network, Optical Network, Data Vortex, Deflection.

1. Introduction

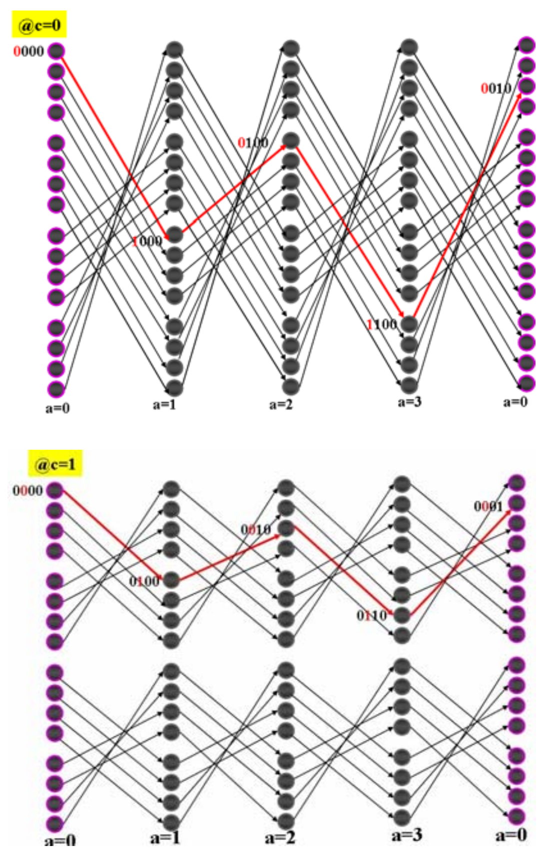
Packet switched interconnection networks are key subsystems in high capacity data communication systems and multi-processor supercomputer systems [2-3]. As I/O ports or high-speed processors that are connected through such networks upgrade dramatically, the interconnection networks must be able to handle very high data rates (tens of Gbit/s) as well as to support a large number of communication ports (on the order of thousands). The key network performance such as throughput and latency must be able to sustain as such networks scale to larger sizes and higher bit rates. A natural way to achieve the higher bandwidth is using optical packet switched interconnections. Current optical fiber and optical amplifier technologies provide enormous operation bandwidth with hundreds of densely packed wavelength division multiplexing (WDM) channels each running at bit rate of tens of Gibit/s. It is thus rather easy to accomplish the high transmission bandwidth in optics. On the other hand, there is still very limited capability in optical processing and optical buffering techniques [4-5]. As a result, the main challenge in these interconnection networks is to handle traffic routing and traffic contention. This has led to difficulty in adapting most existing switching architectures for optical implementations. For example Banyan and Butterfly networks are popular and effective as self-routing electrical switching fabrics networks, however it is very challenging to implement them in the optical domain because of the lack of RAM buffering at each node. Even though pure deflection routing (vs. store and forward routing) is possible, Banyan and Butterfly networks require the deflected packet to travel around the network diameter in order to return to an open path that leads to the target output port. Thus the deflection penalty is prohibitively high and it induces large latency as well as poor network throughput in these two-dimensional network topologies [6].

To take advantages of optics while avoiding extensive buffering and processing optically, Data Vortex network architecture is designed to be a great alternative for the purpose and it is particularly suitable for optical system implementation. The network routing performance has been studied extensively in earlier works and its system implementation and physical layer limitations have also been addressed in a small scale experimental testbed by the research group in Columbia University [7-12]. In this paper, a modified or generalized *k*-ary decoding Data Vortex architecture with multiple header bits decoding stages has been proposed as a potential implementation. The proposal is based on the attractive feature of smaller forwarding hops in the higher *k*-ary decoding Data Vortex networks. Due to different arrangements of routing and forwarding paths, it is essential to study the combined routing performance under different traffic conditions in these extended network architectures.

The organization of the paper is as follows: Section 2 describes the background of the original Data Vortex network design. Section 3 presents the proposed *k*-ary decoding Data Vortex architecture that is extended from the original binary decoding networks. Section 4 presents the details of the performance evaluation through simulation study. Different network and traffic cases are included to thoroughly study the network behaviors as well as the scalability of the results. The comparison study is only focused on between binary and 4-ary networks due to much higher deflection penalty disadvantage in higher *k* systems. The latency performance is also broken down to routing hops, forwarding hops and deflection hops for the analysis and the traffic distribution among cylinder levels within the networks are presented to support our findings. Finally Section 5 concludes and summarizes the study.

2. Background: Original Data Vortex architecture

Data Vortex network is uniquely designed with three dimensional arrangements of the routing nodes. Due to the additional dimension, it allows for bufferless operation using deflection based routing while requiring minimal routing decision that can be implemented electronically within distributed routing nodes. This switching architecture implements a single-packet-routing rule at each node through a traffic control mechanism, and the topology provides multiple open paths to each target address so that deflection routing encounters a much smaller latency penalty (in 2 hops) that is also independent of the network diameter. These network characteristics allow for great network scalability and achieve good throughput and latency performance even for very large network sizes. The bufferless operation offers the simplest possible contention resolution in the optical domain [7-8]. In the physical layer implementation, optical techniques such as dense wavelength division multiplexing (DWDM) are used for achieving ultra high data throughput as well as for simple header bit extraction and decoding. With the available DWDM techniques and the broadband fast switching devices such as Semiconductor Optical Amplifier (SOA), Data Vortex networks allow for relatively short packet (tens of nanoseconds to hundreds of nanoseconds) for efficient operation. This is achieved simply by stacking the data bits along the abundant wavelength channels available within the amplifier bandwidth. Each of the binary header bits uses an additional wavelength channel so that simple and inexpensive filtering and detection can be used for header extraction. More details on physical layer implementation and system limitation can be found in [9-10].



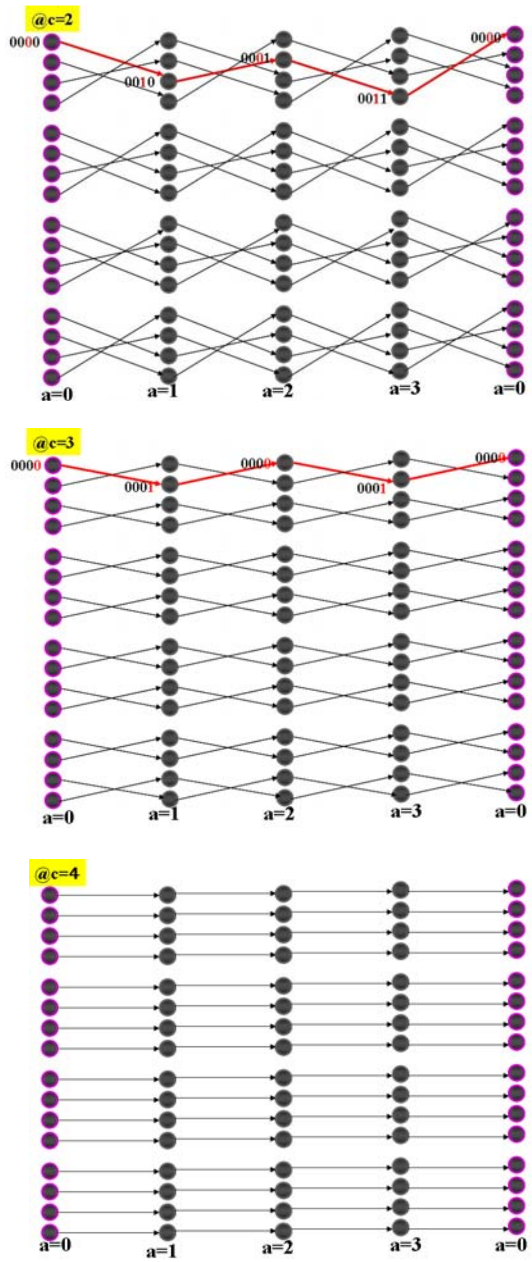


Fig.1. Routing nodes and intra-cylinder links within Data Vortex of *A*=4, *H*=16 and *C*=5

The Data Vortex network can be viewed as a multi-stage interconnection network (MIN). The routing nodes are arranged in concentric cylinders with *A*, *H* and $C = \log_2 H + 1$ designating the number of nodes along the angle, height and cylinder respectively. An example of *A*=4, *H*=16 and *C*=5 Data Vortex network is shown in Fig.1 with proper index of angle (a) cylinder (c) and height (h, shown in binary format) location of the nodes. The intra-cylinder link patterns at each cylinder level are specifically shown from the outermost (*c=0*) level to the innermost (*c=4*) level. These links repeat the same pattern from angle to angle for simple implementation and they route a packet back and forth between two height groups, i.e. with specific (highlighted in red) binary bit alternating between "1" and "0". The inter-cylinder links (not shown in Fig.1) are to forward a packet to an inner neighbor cylinder while maintaining its height location, i.e. a node at angle *a*, cylinder *c* and height of *h* will be connected to an inner node at angle *a+1*, cylinder *c+1* and same height of *h*. Therefore, inter-cylinder paths simply appear to be parallel paths between the cylinders [6]. The network is wrapped around as cylinders, therefore, nodes at angle *a=3* is connected back to nodes at angle *a=0* in a network of *A*=4. The last cylinder maintains the exit height position and it serves as an optical buffering stage in case electrical buffers at the output ports. It is also necessary if angular resolution is required for system implementation when only a specific exit angle is connected to the output ports.

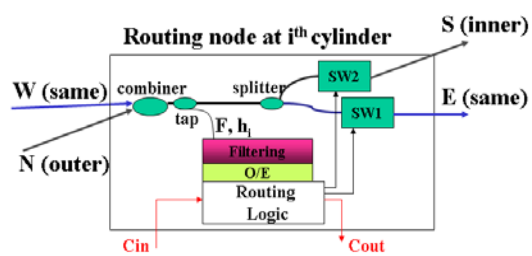


Fig.2 Routing node implementation at ith cylinder

The packet routing in the Data Vortex network is operated in a synchronous and slotted fashion. Packet length is typically chosen to be the same as the hop latency for a simple and efficient implementation. Each node directs a single arrival packet to the next node not only based on the packet's target address, but also based on the inner node's traffic. In Fig.2, a routing node at ith cylinder and its routing logic is shown, and the routing decision is based on the packet frame bit (which tells whether or not a packet is arriving), the corresponding ith header bit (which matches the ith bit of node height or not) as well as the electrical control bit sent from its inner competing node (which permits or blocks the outer traffic). Both the frame and header bits can be extracted by a small power tap and through passive filtering and low packet rate optical/electronic (O/E) conversion. The traffic control signal is generated at each node to properly permit or block the packet of the outer cylinder so that the single-packet-routing rule is always satisfied for bufferless operation for all the routing nodes [11].

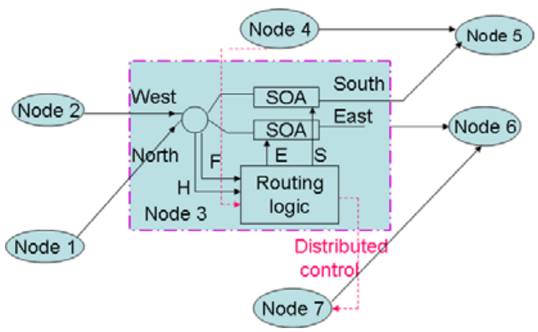


Fig.3 Distributed control signal among routing nodes

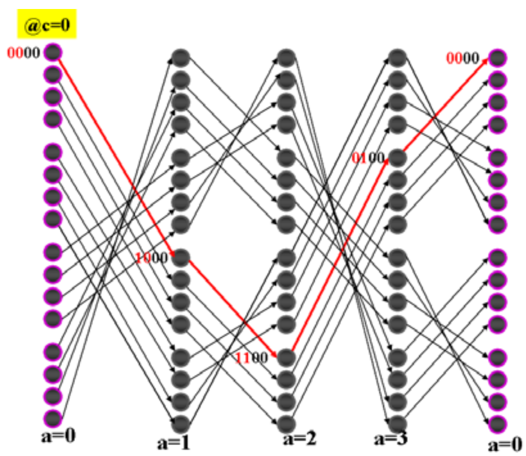
Fig.3 shows the distributed control signals (in dashed line) among the routing nodes on different cylinders. As an example shown, node 3 and node 4 both send packets to node 5, therefore node 4 of the inner cylinder generates a control signal for node 3 to set up the single packet routing rule. Similarly node 3 generates a control signal for its outer cylinder's competing node 7 since they both send packet to node 6. Packets receiving the blocking control are deflected to stay on the current cylinder which acts as virtual buffers with a two hop delay penalty. Once the packet arrives at the correct target, it exits the network in the innermost cylinder. As mentioned, the last cylinder allows for additional optical buffering if necessary. If not all angles at exit are connected to output ports, the last cylinder also deals with angular address resolution of the packet. More details on the angular resolution and choice of angles vs. network height or I/O ports have been discussed in [12].

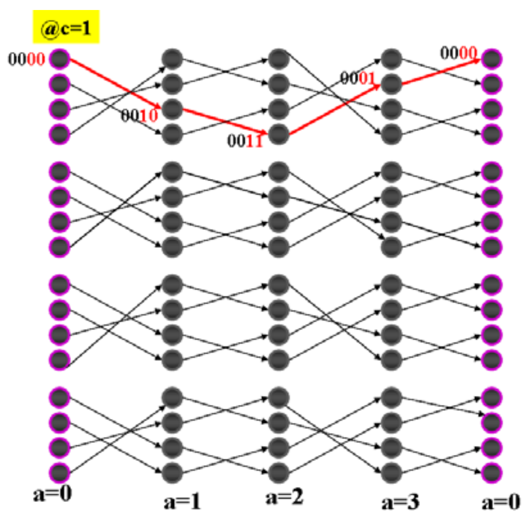
As mentioned above the routing in the Data Vortex network is progressed through a series of binary-tree decoding stages. Each node only decodes a single header bit in the binary target address that corresponds to the node's specific cylinder, and it decides to route the packet further in the current level (current cylinder) or forward the packet to the next level (inner cylinder). Successful forwarding is also dependent on the availability of open control signal due to the traffic condition of the inner cylinder. At different traffic conditions, overall traffic latency is accumulated through routing, deflection and forwarding hops. In this study, we are interested in the optimum arrangement of the routing and forwarding paths. In particular, we explore the potentials of non-binary tree decoding stages.

3. Extended to *k*-ary Data Vortex

Since the header decoding is extremely simple with passive filtering and low speed electronics for detection, it is possible to allow for multiple header bits decoding at each stage. In general, we can implement the Data Vortex network using *k*-ary decoding (i.e. $\log_2 k$ header-bit-decoding) at each routing node, where the binary-tree decoding specially uses k=2. We are interested in exploring alternative implementations of the Data Vortex network as well as verifying the optimum arrangement of the routing and forwarding paths in these networks. Because each packet spends at least C (number of cylinders) hops just forwarding from input port to output port assuming they do not need to stay on the cylinders for additional routing or for traffic contention induced deflection, it is important to study whether or not the overall latency has been minimized under the original Data Vortex network design, or if alternative arrangement of routing and forward paths would change or improve the latency or routing performance. The optimum layout should achieve a best combined routing performance in data throughput and latency. In extending the binary-tree decoding in the Data Vortex network to general *k*-ary decoding at each stage, we maintain the single packet routing condition and the usage of bufferless routing nodes to facilitate the optical implementation of the networks. Therefore, the performance study in this paper is bounded by such design constraints. Future work may address the performance variation in the case of networks with node buffering capabilities, however such networks must require additional hardware cost and implementation complexity [8].

In the case of binary tree decoding in regular Data Vortex networks, each node decodes one header bit, and the routing on a specified cylinder chooses one out of two groups (upper vs. lower group or specific header bit being "1" vs. "0"), and the deflection latency penalty in the network is two hops. If we extend the concept to general *k*-ary decoding, each stage then decodes $(\log_2 k)$ bits, and each hop along the same cylinder allows for the packet to choose one out of k groups (i.e. specific $(\log_2 k)$ bits alternate among all possible k combinations). When the corresponding $(\log_2 k)$ header bits in the target address are matched with the $(\log_2 k)$ bits in the routing node height address, the packet will proceed to the next inner cylinder if the corresponding traffic control opens the path at the same time. So the same traffic control mechanism is set up for the purpose as that in the original Data Vortex network. Otherwise, the packet stays on the current cylinder for further routing or deflection. Since no buffering is necessary the routing logic is kept as minimal as possible. Compared to binary Data Vortex networks, the deflected packet also undergoes longer delay penalty due to the need to go through all k hops to return to the matched height group or to the open path to the next cylinder routing.





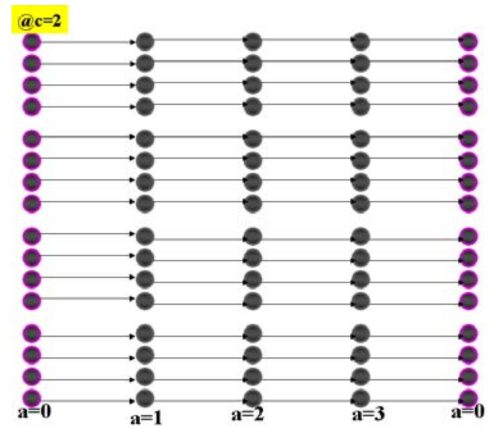


Fig.4 Routing patterns at each of the three cylinders in a 4-ary decoding Data Vortex network. A=4, H=16, $C = \log_4 H + 1 = 3$

In order to allow for a complete permutation of k groups based on the corresponding $(\log_2 k)$ header bits, the intra-cylinder routing paths are slightly modified from the binary-tree decoding networks. In Fig.4 an example of *4*-ary decoding network is shown where each hop decodes 2 header bits in a network of A=4, H=16 and the number of cylinders is $C = \log_4 H + 1 = 3$. Note the interconnection patterns at different cylinders in the binary decoding Data Vortex network are combined and also reversed at proper angles to construct the *4*-ary decoding networks. In such an arrangement, a packet takes k hops to go through all k possible groups along the cylinder, so the deflection latency penalty would increase to 4 hops in *4*-ry decoding networks, which is the obvious disadvantage of using bigger value of k. On the other hand, in extended k-ary Data Vortex implementation, the number of cylinders required is $C = \log_k H + 1$, assuming the last cylinder maintains the same height just for output buffering purpose as that of regular Data Vortex network. As a result, the forwarding latency or number of cylinders is much smaller with a larger value of k. In this study, we choose H so that $\log_k H$ is kept an integer for simplicity. We are specifically interested in *4*-ary networks and its performance comparison with regular binary-decoding Data Vortex network for gain insight of optimum arrangement of routing and forwarding paths. We expect that larger k (k>4) would cause too much deflection latency and traffic backpressure, which will be verified in the 4-ary network study. The node implementation in 4-ary network is slightly modified with the need to filter and detect 2 header bits in parallel instead of a single header bit and it is shown in Fig.5. While the routing node complexity and speed is kept at the same level, the additional filter and detector increase the hardware cost slightly especially when the number of routing nodes are large.

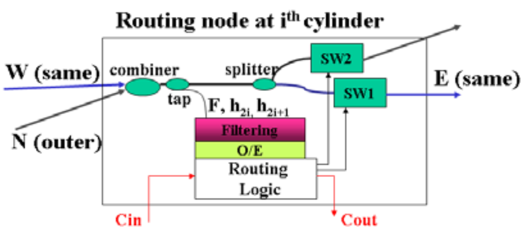


Fig.5 Routing node implementation at i^{th} cylinder in *4*-ary decoding network

4. Performance evaluation

In order to compare the network performance with different *k*-ary decoding schemes and find the optimum arrangement, a C/C++ event simulator is specially developed to evaluate these networks with various operation and traffic conditions. In Data Vortex, once the packets are accepted at injection point, they are routed through the network without loss. Therefore, the measure of through performance is calculated as the rate of successful injection. The latency and latency variation statistics is collected by examining the packets that reach the output ports during the simulation. We particularly focus on comparison between the binary-tree decoding network and a network using *k*=4 decoding scheme due to much more significant deflection latency penalty and traffic backpressure in larger *k* cases.

4.1 Network Cost and Throughput and Latency Performance

To make a fair comparison, the number of input ports is kept the same and the number of routing nodes and routing links that mainly determine the network cost are either same or in the comparable range. First we studied a regular Data Vortex network, network P with *C*=9 and *H*=256. Packets at only injected at a single angle i.e. A_{in}=1 (so number of I/O is the same as the *H*) in a network of *A*=4. We compare its routing performance with two networks Q (*A*=8, *C*=5, *H*=256) and Q' (*A*=7, *C*=5, *H*=256) with *4*-ary decoding both inject using A_{in}=1 to keep the same number of I/O ports as that in network P. In both Q and Q', every stage or cylinder level decodes 2 bits and locates 1 out of 4 height groups. Since the number of cylinders *C*=5 is almost half of that in network P, we allow network Q and Q' to have about twice of the network angles for a similar hardware cost. The detailed hardware comparison is listed in Table 1 below. For the same number of I/O ports, the cost of network P is between that of network Q and network Q'. In this study, we assume no angular resolution is required; therefore, packets that arrive at the correct height will immediately exit the network and be converted to electronic domain. Sufficient electrical buffers are assumed to accept any arrival packet at the output port. If additional angular resolution is required, we shall keep that in mind when we examine the results of different angle networks because a larger angle network generally requires additional hops in the last cylinder before packets exit the optical network.

Table.1 Hardware comparison in network P, Q and Q'

	k=2 Network P	k=4 Network Q	k=4 Network Q''
Number of I/O, H	256	256	256
Number of angles	4	8	7
Number of cylinders	9	5	5
Number of nodes	9216	10240	8960

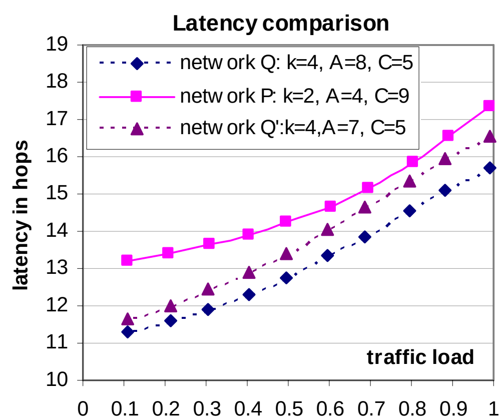


Fig.6 Latency comparison of network P, Q and Q'

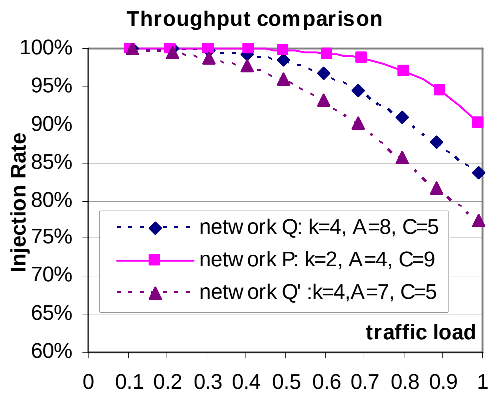


Fig.7 Throughput comparison of network P, Q and Q'

Fig.6 and Fig.7 have shown the latency and throughput performance comparison of the three networks under different traffic loads respectively. For simplicity, all traffics are random and uniform traffic from each I/O port within this study. As shown, overall the regular binary decoding Data Vortex network (in solid line) outperforms the networks with 4-ary decoding. Even though the number of cylinders is much smaller in network Q and Q', for a similar cost, their throughput performances are significantly worse than network P especially at higher load conditions. The average latency of arrival packets is shown to be slightly better in network Q and Q'. However, if angular resolution is required, then network Q and Q' would also encounter more delay in the last cylinder due to the larger A, therefore the gain in latency is not necessarily noticeable.

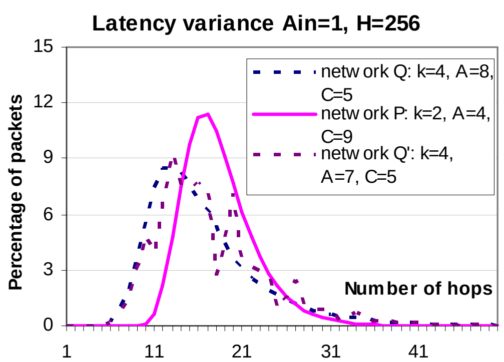


Fig.8 Latency variance in network P, Q and Q' when load=1.0.

The latency distribution in these three networks is shown in Fig.8 for a case of fully loaded condition, i.e. the applied traffic load=1.0. The results have indicated that network Q and Q' both have pushed part of the packets to much shorter latency (left side of the distribution curve moves earlier), however due to the larger deflection delay which also induces more traffic backpressure, the overall distribution has wider deviation range in these *4*-ary networks, and its tail also extends noticeably further even though the percentage of packets with very large latency is kept small. In comparison, the binary tree decoding Data Vortex has a narrow and confined distribution curve.

4.2 Latency Performance in breakdown categories

Next, we studied a case where two networks chosen have the exact same hardware cost for the given same number of I/O ports. Network M in binary tree decoding Data Vortex has *A*=5, *C*=9 and *H*=256 whereas network N using *4*-ary decoding stages has *A*=9, *C*=5 and *H*=256. Both networks use a single angle injection A_{in}=1 for the simple comparison. The detailed hardware comparison is shown in Table 2 below. The throughput and latency performance under different traffic loads are shown in Fig.9 and Fig.10 respectively. In the average latency plot in Fig.10, we also plot individual category of delay such as average deflection hops and average routing hops to gain further insights. The forwarding number of hops is not shown but it is fixed to the number of cylinders. As seen, it verifies the better throughput performance in binary Data Vortex network especially at heavier traffic load conditions given the same network cost. The average number of hops in network N is slightly better, however keep in mind it may experience additional hops in angular resolution if compared to that in network M. Fig.10 also shows why it doesn't gain much advantage in latency performance in *4*-ry network even though its forwarding hops (C=5) is 4 hops smaller than that in the binary network (C=9). The number of the routing hops on average is shown to be about 3~4 hops more in network N compared to that in network M. In this case, the deflection hop only counts those hops of staying in the current cylinder due to unavailability of open control signal, and the penalty hops are lumped to the routing hops. Therefore, the results show that the deflection probability is pretty close in two networks under different traffic conditions, however the routing hops is much larger in network N due to larger deflection hop penalty and generally more hops required to match the permutations even in regular routings.

Table.2 Hardware in network M and N

	k=2 Network M	*k*=4 Network N
Number of I/O, *H*	256	256
Number of angles	5	9
Number of cylinders	9	5
Number of nodes	11520	11520

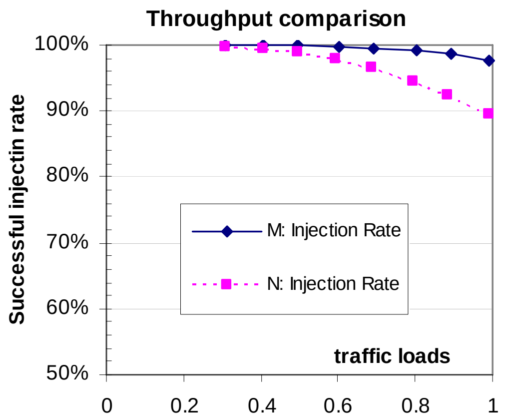


Fig.9 Throughput comparison in network M and N.

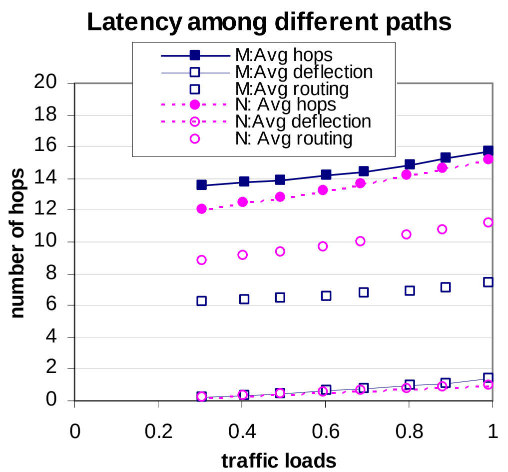


Fig.10 Latency comparison in network M and N

Fig.11 shows the latency distribution curves of the two networks in the case of load=1.0. Similar to the results in Fig.8, it is found that certain packets go through a smaller number of hops leading to the earlier front trail of the distribution curve, however the overall distribution is wider and the ending tail is longer as a result in the *4*-ry network. On average the routing hops in network N is larger than that in network M, and this cancels out the advantage of less forwarding hops. The latency and the latency distribution performance are also closely related to the throughput performance because of the backpressure effect in traffic. If more packets are pushed through the network in a faster pace, it allows for better throughput, otherwise, the packet occupies the network resource which causes additional deflection and delay. The statistics of all the packets within the network prove that *4*-ary routing does not bring sufficient benefit in latency and throughput performance even with a much smaller number of the forwarding hops.

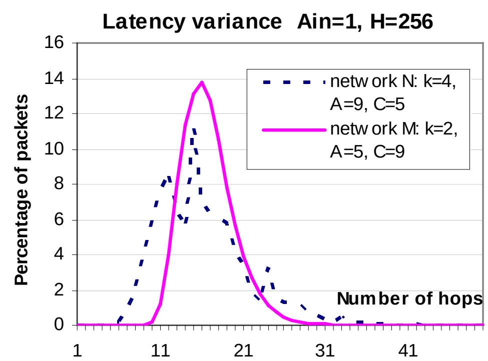


Fig.11 Latency variation at load=1.0 in network M and N

It is also important to study the performance comparison for different network sizes. For this purpose, we used two additional network M_2 (*A*=6, *C*=11, *H*=1024) and N_2 (*A*=11, *C*=6, *H*=1024) both with A_{in}=1. These two networks are chosen because they share the same hardware cost for the same number of I/O ports while support much larger network height or I/O numbers. The detailed hardware comparison is shown in Table 3.

Table.3 Hardware in network M_2 and N_2

	k=2 Network M_2	*k*=4 Network N_2
Number of I/O, *H*	1024	1024
Number of angles	6	11
Number of cylinders	11	6
Number of nodes	67584	67584

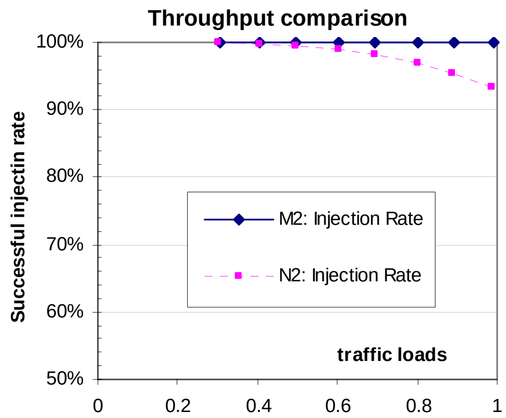


Fig.12 Throughput comparison in network M_2 and N_2

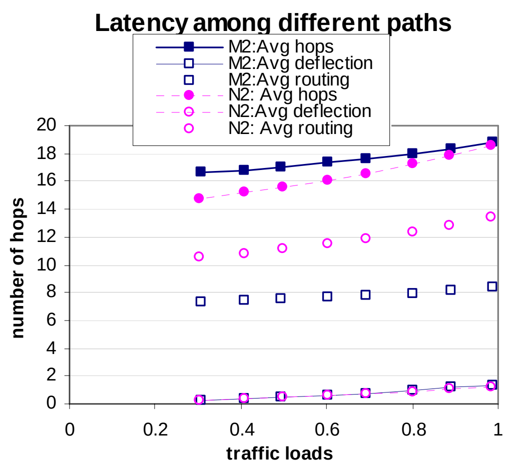


Fig.13 Latency comparison in network M_2 and N_2

The routing performances in throughput and latency of M_2 and N_2 are compared in Fig.12 and Fig.13 respectively. As shown, the performance difference between these two networks follows a very similar trend as that in the comparisons of network M and N. The results have confirmed that at different network sizes and network load conditions, the binary decoding Data Vortex network almost always outperforms the *4*-ary network, mainly due to its inherent small deflection latency penalty and more frequent encounter of open paths between different cylinder levels. Therefore, binary decoding Data Vortex provides the best combined routing performance with the optimum routing and forwarding paths arrangement for bufferless operations.

In addition to the overall routing performance, the latency distribution curves also show some non smoothness in *4*-ry decoding networks which is not present in regular Data Vortex network. It seems to be dependent on the angle of the network. To study this further, several networks with H=256 and a varying network angle are compared and shown in Fig.14, and all of them use *4*-ary decoding stages with a single angle injection, i.e. A_{in}=1. The distribution curves show that for the fixed number of I/O ports (fixed H and A_{in}), a larger network angle result in earlier leading edge of the distribution curve due to relatively more redundancy in the network resource. It is also shown that at angles that are integer multiple of k such as A=8 and A=12, the distribution curve is rather smooth because of regular distribution of the traffic pattern and equal probability to each node. On the other hand, if A is not an integer multiple of k, traffic may not be evenly distributed among groups of nodes, and this leads to multiple peaks in the distribution curves or rather non-smooth distribution. Only when the number of angles A is relatively large, such non-smoothness becomes insignificant due to contributed hardware redundancy. In comparison, in the binary decoding Data Vortex network, the latency distribution smoothness is rather insensitive to the number of network angles whether A is even or odd.

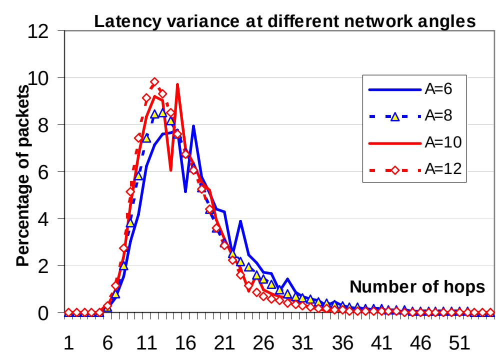


Fig.14 Smoothness of latency distribution curve at different network angles

4.3 Traffic distribution within different cylinder levels

We can gain further insight of the network behaviors by examining the traffic pressure among the cylinders. To compare the two different architectures with different routing and forwarding path arrangements, we record the traffic load or packet count of each specific cylinder at different operation conditions. Packets on the specific cylinder and the ones entering the cylinders are counted as the packet of the cylinder, and each cylinder's packet count is monitored for comparison study.

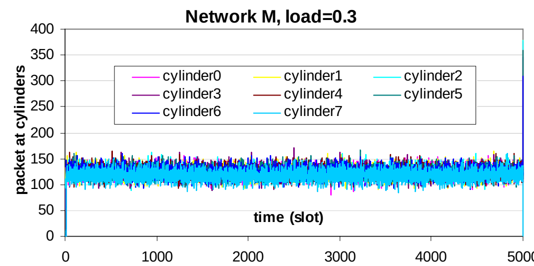


Fig. 15 Packet count of each cylinder in network M with load of 0.3 during simulation

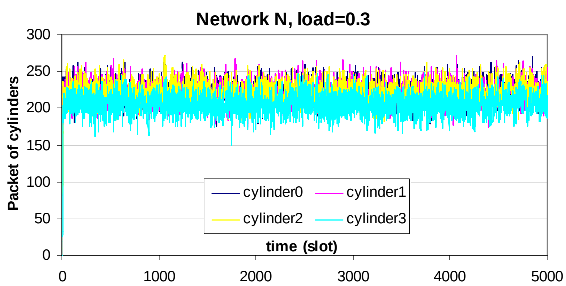


Fig.16 Packet count of each cylinder in network N with load of 0.3 during simulation

We specifically compared the traffic or packet count at each cylinder level in both lightly loaded and heavily loaded conditions. In Fig. 15 and Fig. 16, the packet counts over simulation time (5000 time slots) under traffic load of 0.3 in network M and network N are shown for comparison. To get better view, the ranges of the packet / traffic load (once the traffic reaches a relatively steady state after the initial packet injection) at each cylinder are also shown in Fig. 17 and Fig. 18 respectively. We found that under this lightly loaded condition, both networks distribute their traffic among different levels pretty evenly, and there is only slight difference between outer cylinders and inner cylinders, which indicate no significant traffic back pressure buildups in both network M and network N. In our study, since no angular resolution is required, the last cylinder's packet count is not shown due to immediate exit at the stage. The absolute level of packet count in two networks may not provide a direct comparison due to different number of cylinders, but the pattern and difference between different cylinder levels are compared and focus of the study.

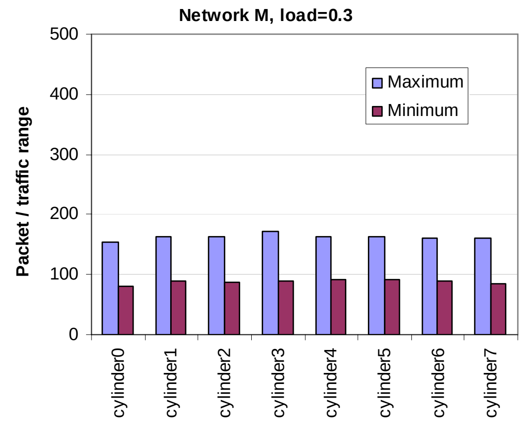


Fig.17 Traffic range at each cylinder in network M with load of 0.3 during network operation

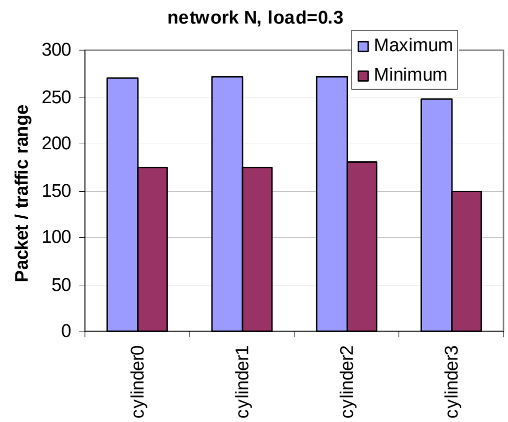


Fig.18 Traffic range at each cylinder in network N with load of 0.3 during network operation

The same networks M and N under a heavier load of 0.8 are also studied for the comparison purpose. The results of the traffic distribution among different cylinder levels are shown in Fig. 19, Fig. 20, Fig. 21 and Fig. 22 respectively. We observed that in network M the outer cylinders bear very similar level of traffic loads, and only the last few inner cylinders carry slightly less loads (more visible compared with lightly loaded condition). On the other hand, in network N, while the total number of cylinders is much less, and outermost cylinder carries significantly more traffic compared with that of the inner cylinders. The difference in each cylinder is much more visible compared to that in network M, and such difference is also more significant in this heavier load condition than that in the lightly loaded network. There are a couple factors contributed to the difference. First of all, network N has about double the angle (for the same hardware cost), which amplify the traffic load difference for each cylinder by a factor of 2 given the same height *H* in both networks. The second factor is due to more accumulated traffic backpressure at the outer cylinders in network N than that in network M. Because of longer deflection penalty and generally longer routing steps at each cylinder in *4*-ary networks, packets are staying in the cylinders for statistically longer period of time. So in comparison, the traffic backpressure is less in binary Data Vortex network, which creates much evenly distributed traffic among the different cylinder levels. The most inner cylinders always carry slightly lower loads compared to their outer cylinders because its input traffic is rather balanced or smoothed after the outer cylinder's routings. The traffic distribution among cylinders explains the overall more effective routing from the binary Data Vortex network, and thus it explains its higher throughput performance compared to its *4*-ary counterpart with the same I/O port and same hardware cost.

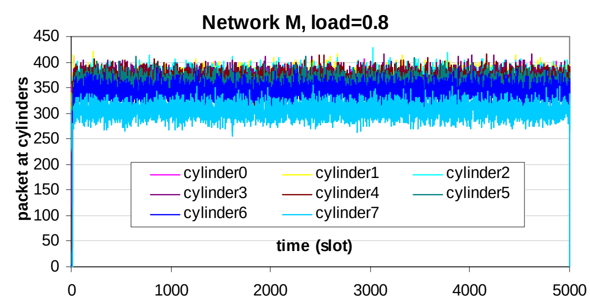


Fig. 19 Packet count of each cylinder in network M with load of 0.8 during the simulation

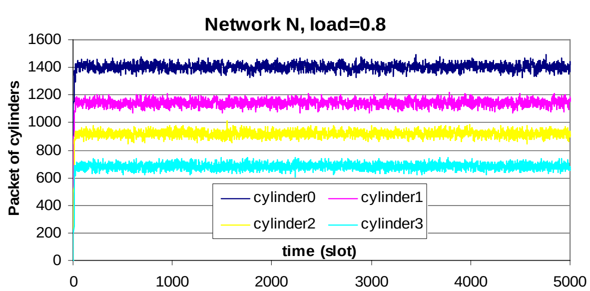


Fig.20 Packet count of each cylinder in network N with load of 0.8 during the simulation

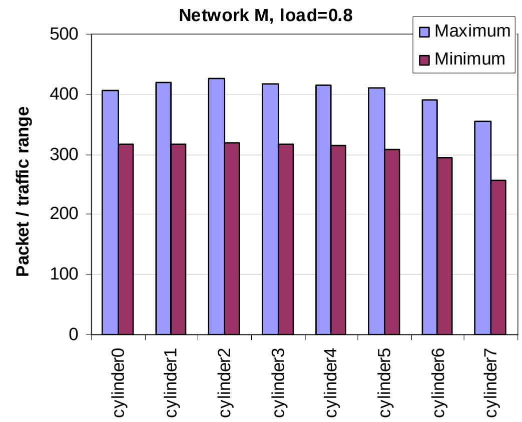


Fig.21 Traffic range at each cylinder in network M during network operation

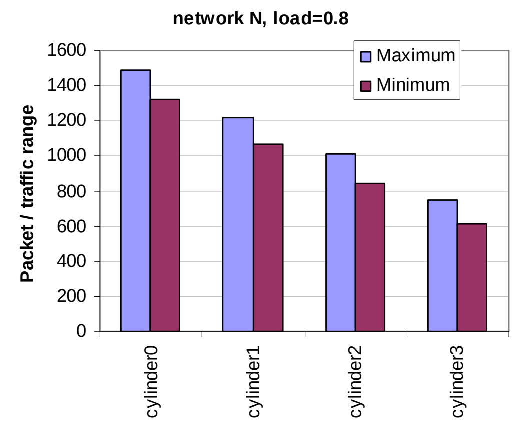


Fig.22 Traffic range at each cylinder in network N during network operation

5. Conclusion

We have explored the potential of general *k*-ary decoding scheme in the Data Vortex networks. The comparison study has focused on studying the network behavior difference between the *4*-ary decoding network and the regular binary-tree decoding Data Vortex network. The results have concluded that overall routing performance is optimum with the original binary Data Vortex networks even though its forwarding latency is much longer than that in a *4*-ary network of a similar network cost. The main reason is that binary decoding provides the lowest deflection latency penalty, which in turn reduces the traffic backpressure during bufferless deflection based operation. Therefore, without any additional buffering at node, the binary Data Vortex networks allow for the system to push the most packet traffic through at the lowest average latency. Future work may explore the effect of buffering capability within the nodes, however additional hardware cost must be included in the consideration.

6. Reference:

[1] Qimin Yang, “Optimum routing and forwarding path arrangement for bufferless Data Vortex Networks”, *The Seventh International Conference on Networking (ICN 2008)*, Mexico, April 13-18, 2008.

[2] H. Jonathan Chao and Ben Liu, “High Performance Switches and Routers”, *Wiley-Interscience*, ISBN 0470053674, 2007.

[3] Gorgios I. Papadimitriou, Chrisoula Papazoglou, Andreas S. Pomportsis, “Optical Switching: Switch Fabrics, Techniques and Architectures”, *Journal of Lightwave Technology*, Vol.21, No. 2, pp.384-405, Feb 2003.

[4] Haijun Yang and S.J. Ben Yoo, “All-Optical Variable Buffering Strategies and Switch Fabric Architecture for Future All-Optical Data Routers”, *Journal of Lightwave Technology*, Vol. 23, No.10, pp.3321-3330, Oct 2005.

[5] Chowdhury, A. Yong-Kee Yeo, Jianjun Yu, Gee-Kung Chang, “DWDM reconfigurable opticaldelay buffer for optical packet switched networks”, *IEEE Photonics Technology letters*, Vol. 18, No.10, pp.1176-1178, May 2006.

[6] B.E.Swekla and R.I.Macdonald, “Tandem Banyan Switching Fabric with Dilation”, *Electronics Letters*, Vol.27, No.19, pp.1770-1772, 1991.

[7] C. Hawkins, B.A. Small, D.S. Wills, K. Bergman, “The Data Vortex, an All Optical Path Multicomputer Interconnection Network”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No.3, pp. 409-420, March 2007.

[8] Assaf Shacham and Keren Bergman, “On contention resolution in the data vortex optical interconnection networks”, *Journal of Optical Networking*, Vol.6, pp.777-788, 2007.

[9] A. Shacham, B.A.Small, O.Liboiron-Ladouceur, K. Bergman, “A fully implemented 12x12 data vortex optical packet switching interconnection networks”, *Journal of Lightwave Technology*, vol.23, pp.3066-3075, 2005.

[10] O. Liboiron-Ladouceur, B.A. Small, K. Bergman, "Physical Layer Scalability of WDM Optical Packet Interconnection Networks," *Journal of Lightwave Technology*, Vol. 24, No.1, pp. 262-270, Jan 2006.

[11] Qimin Yang and Keren Bergman, “Traffic Control and WDM Routing in the Data Vortex Packet Switch”, IEEE Photonics Technology Letters, Vol. 14, No.2, pp. 236-238, Feb 2002.

[12] Cory Hawkins and D.Scott Wills, “Impact of Number of Angles on the Performance of the Data Vortex Optical Interconnection Networks”, *Journal of Lightwave Technology*, Vol.24, pp.3288-3294, 2006.