Received: 4 July 2019 | Revised: 30 September 2019 | Accepted: 20 January 2020

DOI: 10.4218/etrij.2019-0318

ORIGINAL ARTICLE

ETRI Journal WILEY

Performance analysis of torus optical interconnect with data center traffic

Abhilasha Sharma | **Sangeetha Rengachary Gopalan**

School of Electronics Engineering, VIT University, Chennai, India

Correspondence
Sangeetha Rengachary Gopalan, School of Electronics Engineering, VIT University, Chennai, Tamilnadu, India.
Email: sangeetha.rg@vit.ac.in

Two-dimensional torus network nodes are typically interconnected using *XY* routing algorithm for transmitting a packet from a source node to a destination node. In *XY* routing, if all the paths are used efficiently, the throughput and latency can be improved. In this paper, to utilize all the paths efficiently, we propose a novel binary optical routing algorithm (BORA) to improve the throughput and latency. The throughput is calculated according to the injection rate and number of packets received at the destination. The *XY* routing algorithm and proposed BORA are implemented using objective modular network testbed in C++ simulation software and the results are analyzed and compared. In this paper, the simulation results show that the network latency reduces to 50% while using the proposed algorithm; moreover, the throughput is also improved.

KEYWORDS

binary routing algorithm, latency, throughput

1 | INTRODUCTION

Data centers (DCs) are interconnected across the world through routing and transport technologies to provide a pool of resources [1]. In addition to cloud services, DCs provide many other applications, such as social networking, web searching, email, and online gaming [2]. All these services are provided by shared DCs because of the high cost of the physical equipment and their maintenance. DCs consist of thousands and thousands of server racks. These server racks communicate through an intra-DC network. The role of DC networks (intra- and inter-DC networks) is to effectively organize the data (or requests) to deliver services with strong reliability [3,4] and excellent performance to users. The entire performance of the DC networks depends on the performance of optical interconnects. The exponential increase of traffic in cloud services and internet [5] has increased the requirement for high-performance, scalable optical interconnects. Hence, many DC networks were proposed in the literature. The DC networks include hybrid electrical/optical structures (eg, Helios [6]), direct networks (eg, CamCube [7]), and server-centric networks (eg, Benes [8]). To fully investigate all the network capacities and parameters, a series of DC network routing algorithms were proposed in the literature. The crucial challenges to be addressed in any DC network are throughput, latency, fault tolerance, and reliability. The routing scheme in a network is responsible for addressing the aforementioned crucial challenges. An efficient routing scheme provides redundant paths. The network latency, throughput, fault tolerance, and reliability of the network can be improved when the packets are routed using these redundant paths.

Efficient topologies and routing algorithms [9–14] are required to manage the increase in traffic within the optical interconnects of DC networks. A non-blocking Spanke-type scalable network with distributed control was proposed in [8]. This architecture was based on wavelength-division multiplexing and optical packet switching; moreover, it was analyzed for up to 4096 ports in [8]. A bidirectional interconnection network was proposed in [15–19], where an efficient routing algorithm was used to transmit the packets in both the directions. This architecture also exhibited the features of low latency, high throughput, and scalability. Torus topology is a bidirectional interconnected network [20–24], which is also a suitable candidate for intra-DC networks.

1.1 | Torus topology

Torus is a direct network topology in which every node serves as an input link, output link, and a switching node in the network [25]. There are several reasons for considering the torus network topology as a suitable candidate for optical interconnects, which are discussed subsequently. The torus topology is regular (ie, all nodes have the same degree pertaining to its neighbors) and edge symmetric, which help to improve load balance [25]. This topology also presents high path diversity owing to its symmetric nature. The property of path diversity is used for maintaining multiple distinct nonoverlapping paths between each node pair to serve the purpose of routing and managing the traffic [26]. The high intra-DC network traffic can be balanced by distributing it over the redundant paths available (using the property of path diversity). This topology is highly scalable, reliable, and supports bidirectional signaling.

The total number of nodes in a two-dimensional (2D) torus topology is calculated using (1), where N is the number of nodes in each dimension.

$$X = N^2. \tag{1}$$

The total number of nonoverlapping paths in a 2D torus network topology is calculated using (2), where δ_j is the number of hops from node 1 to node 2 [25].

$$P_{\mathrm{sd}} = \frac{\left(\sum_{i=0}^{1} \delta_j\right)!}{\prod_{i=0}^{1} \delta_j!}. \tag{2}$$

For the sake of analysis, a 2D torus topology of network size 4 × 4 is shown in Figure 1. In a 4 × 4 torus network, the number of nodes in the X or Y dimension is equal to four (as depicted in Figure 1). Therefore, the total number of nodes is equal to 16 (using (1)). Assuming, node 1 is the source node and node 2 is the destination node, the total number of hops to reach from the source to the destination (δ_j) is four, as shown in Figure 1. Then, the minimum number of nonoverlapping paths is 70, as calculated using (2).

Traditionally, the XY routing algorithm is implemented in the torus network topology. The torus topology has high network performance in terms of low latency and high throughput, as described in [27]. Path diversity is an important property of the torus topology as it provides multiple nonoverlapping paths. If these paths are utilized properly, latency and throughput can be improved.

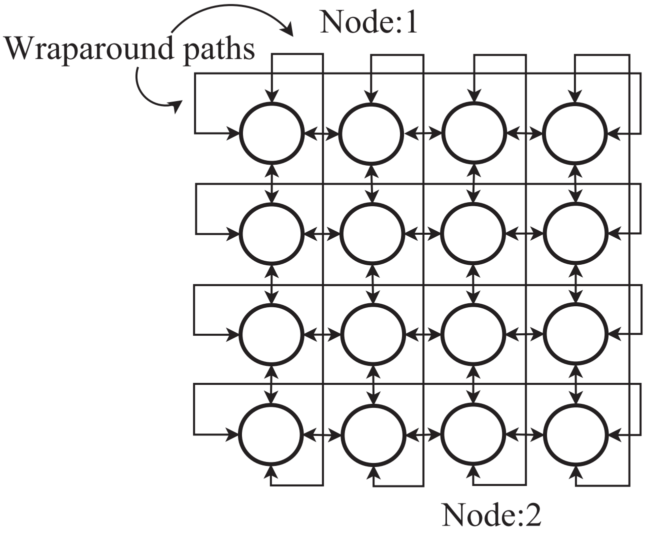

FIGURE 1 Architecture of 4 × 4 torus topology

In the XY routing algorithm, the paths that directly connect the end nodes are utilized for deflection routing and inherent buffering [16]. In this paper, we proposed a binary optical routing algorithm (BORA), which provides low latency without compromising the property of inherent buffering of the torus network topology. The network was simulated for a network size of $N \times N$, where $N = 8$, 16, 32, and 64. The BORA provides improved performance in terms of low latency and high throughput (analyzed in terms of the injection rate and number of packets received).

The rest of the paper is organized as follows. Section 2 describes the XY routing algorithm. The proposed routing algorithm is explained in Section 3. Section 4 describes the properties of the proposed routing algorithm. The DC traffic generation and simulation setup are described in Section 5. Performance analysis and simulation results are described in Section 6; moreover, Section 7 concludes the paper.

2 | *XY* ROUTING ALGORITHM IN TORUS NETWORK TOPOLOGY

In the XY routing algorithm, there are two routing paths (pertaining to the $X+$ and $Y+$ ports) available for routing. The XY routing algorithm utilizes the wraparound paths only for inherent buffering. If the routing paths are not available, then the packet is inherently buffered in the wraparound path. When both the output links are available, then the wraparound path is idle (free). If the wraparound paths are effectively used in the idle time slots, the latency and throughput of the network can be improved. In the BORA, there are three available routing paths (two output paths and one wraparound path [whenever it is idle]). The wraparound path is also used as the output path during the idle time slots without compromising its buffering capacity. The packet is also inherently buffered to the wraparound path when the routing paths are not free. Even when both the output links are free, if buffering is not required, then these idle wraparound paths are used to route the packet. In Ref. [27], the *XY* routing algorithm was used for the torus network topology, where each node was a 2 × 2 switch, with two input links (*X*− and *Y*−) and two output links (*X*+ and *Y*+), as shown in Figure 2. In the *XY* routing algorithm, only one packet is processed in a single time slot. The packet is composed of two parts: payload and header. All the switching nodes are connected with both electrical and optical links. The payload traverses on the optical links, which are of the same length. The header traverses on the electrical links, which are shorter than the optical links [28]. Owing to the unequal lengths of the links, the header reaches the node earlier for header processing. For further routing of the packets, the state of the switches is decided by the header bits after electronic signal processing. Meanwhile, the optical payload arrives at the node and is routed to the desired destination. For example, the *XY* routing algorithm for the torus network of size 4 × 4 is shown in Figure 2. Here, we assume that (0, 1) is the source node and (3, 2) is the destination node; further, each node has two output links and one wraparound path. Out of the two output links (*X*+ connected to node (0, 2) and *Y*+ connected to node (1, 1)), high priority is set for the *X*+ output link. Therefore, if a packet is moving from node (0, 1), based on the highest priority, it will be routed to the node connected with the *X*+ output link. If that output link is not free, the packet will be routed to the wraparound path (using *Y*+) for buffering. If the packet is routed by the *Y*+ link, then with the help of a deflection routing scheme, the packet is inherently buffered within the network until the output link *X*+ is free [27]. In this scenario, the *Y*− link is also available; if this path is utilized properly when it is idle, the network performance can be considerably improved. While using the *X*+ output link, the packet is routed through nodes (0, 1) ⇒ (0, 2) ⇒ (1, 2) ⇒ (2, 2) ⇒ (3, 2), requiring at least four hops through the path. If *X*+ is not free, then the packet utilizes the *Y*+ output link. Now, the packet is routed through (1, 1) ⇒ (2, 1) ⇒ (3, 1) ⇒ (3, 2), which is four hops. Instead, if the packet is transmitted through the wraparound path through *Y*− output link, then it reaches in two hops ((3, 1) ⇒ (3,2)). This can be achieved with proper binary signaling. In this paper, we propose the BORA to utilize wraparound paths to reduce the latency.

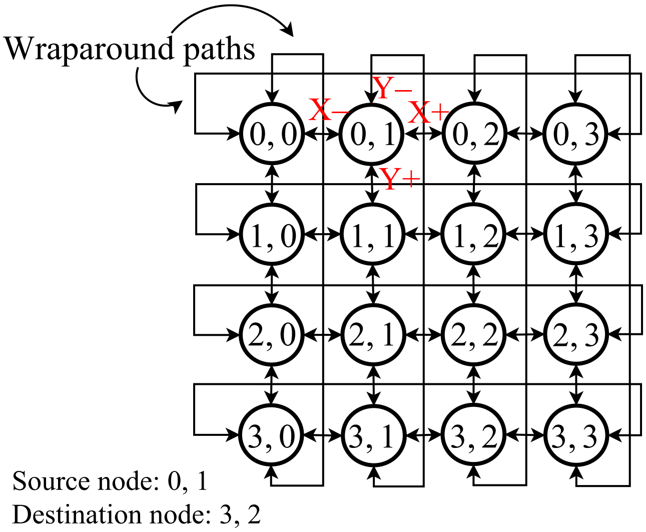

FIGURE 2 *XY* routing algorithm implemented in a two-dimensional (2D) torus topology with network size of 4 × 4

3 | BINARY OPTICAL ROUTING ALGORITHM

3.1 | BORA: Addressing scheme

In BORA, each node address represents an *n*-digit radix-*r* (where "*r*" is the number of nodes in one dimension) gray code address. In the 2D torus network topology, the total number of nodes is found using (1); moreover, every node is connected by a pair of bidirectional links. This type of node addressing scheme is possible owing to the following properties of gray codes:

- *Unit Distance Property*: By implementing the node addresses in 2D binary gray codes, the neighboring nodes differ by one bit and the subsequent neighboring nodes differ by two bits.
- *Cyclic Property*: The difference in the node addresses of the first and last rows and columns is equal to one bit. Owing to this property of gray code, each node has four neighboring nodes. This property provides end-to-end connectivity by enabling wraparound paths.

With the help of these two important properties of 2D binary gray codes, a routing algorithm for this network topology was developed. Moreover, in this routing algorithm, we utilized the wraparound paths. A simple 4 × 4 torus network topology is shown in Figure 3. In this figure, the effect of the properties of the gray code is indicated. For example, nodes (0001, 0010, 0100, 1000) are the neighbors of the node (0000) owing to the cyclic property. All these node addresses have one bit difference with respect to node (0000). Similarly, the node addresses of nodes (0100) and (0101) differ by one bit as per the unit distance property.

3.2 | BORA: Routing scheme

The data transmission is divided into "*n*" time slots. Each packet is assumed to have a different length and

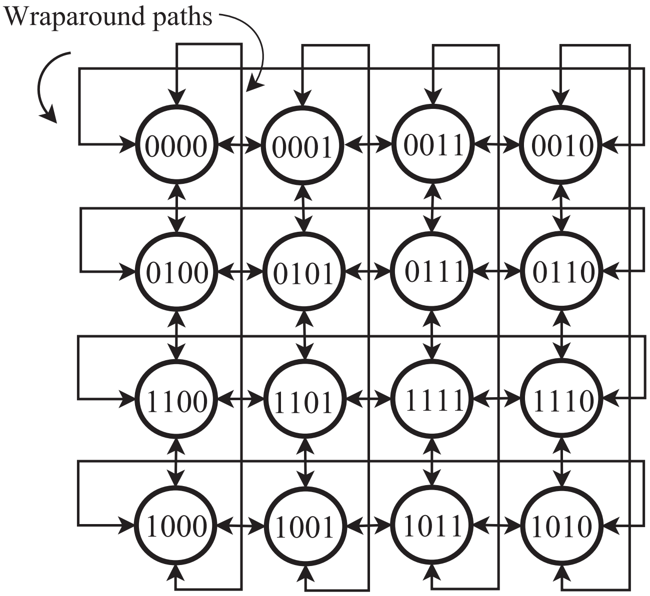

FIGURE 3 Binary addressing in 2D torus topology with a network size of 4 × 4

is synchronized with the time slots. At every node, only one packet is processed in a single time slot. Assuming, $N = (T, C)$ as a 2D torus network, where T is the set of nodes numbered from 0 to $2^n - 1$ (for a network size of $N \times N$) and C is the set of bidirectional links. Each node is connected to the neighboring nodes with a one bit difference in their node addresses. Let us assume that the hamming distance is represented as $H_D\,(x, y)$ between two nodes x and y. The difference in the number of bits between the source and destination node addresses is defined by the hamming distance.

Let us assume that $u\,(i)$ represents the bit positions at which the source and destination node addresses differ by one bit. The priority of the output ports is indicated by $u\,(i)$. The first set of neighboring nodes is denoted by $H_S\,(x, r)$, which is a hamming shell of radius r centered at node x. It is defined as the set of nodes $\{y | y \in T \forall H_D\,(x, y) = r\}$, which is the set of neighboring nodes at a radius of r. The value of r for the first neighbors is equal to one, which implies that only these nodes contribute to the formation of the hamming shell whose node addresses differ by one bit from the node address of node x. The secondary set of neighboring nodes and the remaining neighboring nodes contribute to the formation of a hamming ball of radius r, which is denoted by $H_B\,(x;r)$. The hamming ball is defined as the set of nodes $\{y | y \in T \forall H_B\,(x, y) \geq r\}$ that has a node address that differ by two or more bits from the node address of node x. The desired output link for routing the packet is identified using a priority scheme based on the minimum hamming weight $\left(W\left(H_B\right)\right)$ of the neighboring nodes. $W\left(H_B\right)$ denotes the hamming weight of the neighboring nodes. Out of the two selected nodes connected to the port with high priority, the node with the minimum hamming weight is given the priority to route the packet. The pseudocode for the routing algorithm is explained in Algorithm 1.

Algorithm 1 Binary Optical Routing Algorithm

```
$y$: destination node and $x$: current node;
Initially, $next\_hop$: Optimized;
data_packet $d$ : $(y, next\_hop, message)$;
$next\_hop = Optimized (= W(H_B)_{\min})$;
if $H_D(x, y) == 0$ then
    Data packet $d$ has reached its destination.
else
    Begin:
    $r := H_D(x, y)$;
    $L_F := L_{F1}, ..., F_r$; set of forward links
      $L_R := L_{R1}, ..., L_{n-r}$; set of backward links
    while $next\_hop = Optimized$ do
        if $next\_hop == Optimized$ then
            If all paths are not faulty in $L_F$ THEN
send $d$ adaptively along a non-faulty forward link;
        else
             select a path to a node $u \in H_S(y, r+1)$ along
a non-faulty link $L_{Ri} \in L_R$ such that a non- faulty path
$L_{Fi}, L_{Fj} \in L_F \cup L_{Ri}$ from $u$ to a node $v \in H_S(y, r-1)$;
        end if
    end while
end if
```

For example, in Figure 4, we assume that (0001) and (1011) are the source and destination nodes, respectively. The first neighboring nodes form a hamming shell, that is, $H_S\,(x, r) = \{0000, 0011, 1001, 0101\}$; it is to be noted that all these node addresses have a one bit difference from the source node (0001). Similarly, the hamming ball is formed by the secondary neighbors, which are the node addresses that differ in two bits from the source node. Among the four neighboring nodes, the priority has to be fixed for transmitting the packet to the desired destination node. This is performed by using the difference in the bit positions of source and destination addresses. In this example, the source and destination nodes differ in the first and third bit positions. The source and destination addresses differ at the underlined bit position numbers, that is, *SourceNode*:$\underline{0}0\underline{0}1$ and *DestinationNode*:$\underline{1}0\underline{1}1$ (first and third positions). Hence, $u\,(i) = \{1, 3\}$. Therefore, ports 1 and 3 are used to route the packet with the minimum number of hops. Out of these two ports, the packet is routed to the node with less hamming weight. In this case, the two neighboring nodes are of equal hamming weight.

Therefore, one of the output nodes is selected with alternate priority. Therefore, the packet is transmitted to node (0011) and if it is not free, then the packet is forwarded to node (1001). A similar arrangement is created for all nodes for transmitting the packet to the desired destination. The

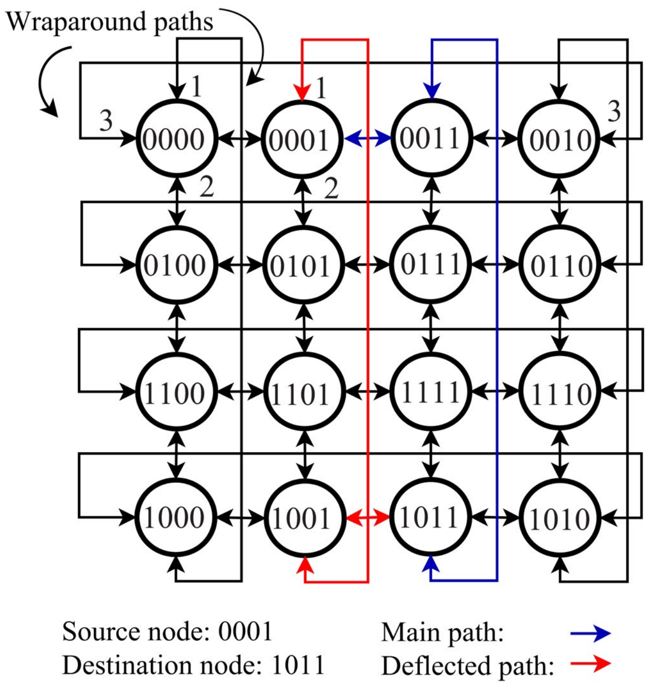

FIGURE 4 Binary optical routing algorithm (BORA) implemented in a 2D torus topology with a network size of 4 × 4

parameters of the routing algorithm implemented for this example are listed in Table 1.

4 | PROPERTIES OF BORA

The properties of BORA can be generalized for any bidirectional multistage interconnection network.

Theorem 1 *The packets cannot be lost or blocked.*

Proof Let us assume that the source node and destination node as x and y, respectively. In the torus topology, each node has four neighboring nodes. Each node has a set of nodes that forms a hamming shell and hamming ball. The hamming shell $\left(H_S(x, r)\right)$ for the node x is the set of neighboring nodes whose addresses differ in one bit from that of node x. Similarly, the hamming ball $\left(H_B(x, r)\right)$ for node x is formed by the secondary set of neighbors whose addresses differ in two bits from that of node x. Among the four neighboring nodes, the priority has to be fixed for transmitting the packet to the desired destination node. This is performed by considering the difference in the bit positions of source and destination addresses. These bit positions designate the ports that are used to route the packet with the minimum number of hops. Out of these ports, the packet is routed to the node with less hamming weight.

If that particular selected link/node is not free, then the packet will be routed to the next neighboring node with a minimum hamming weight of $W(H_B)_{\min}$. In this manner, all the three routing paths are available to route the packets without compromising the buffering capacity. Therefore, the packets cannot be lost in either the link or the nodes.

Hence, the packets will be forwarded until they reach their destination. The subsequently presented Lemma 1 and Lemma 2 provide further proof for the theorem. □

TABLE 1 Parameters and their values

Parameters	Values
Source address (x)	0001
Destination address (y)	1011
$H_D(x,y)$	2
$u(i)$	{1,3}
$H_S(x,r)$	{0000, 0011, 0101, 1001}
$W(H_B)_{\min}$	{0011, 1001} any of the two nodes with alternate priority.

Lemma 1 *The destination node* y *is reachable from node* x *even if* m *busy links (out of total* f *links) are present at node* x*; moreover,* $r = H_D(x,y) \geq 1$*, with respect to node* x*. Then, the packet is not dropped in the nodes or links.*

Proof Let us assume that node x has m busy links (such that $r \leq m < f$); however, the data packet d is not dropped. This is because of the high path diversity and availability of three routing paths, which do not compromise the inherent buffering capacity; further, $f - m$ number of free links are still available. The packet d will be routed to one of the remaining neighboring nodes b, which belongs to $H_S(y, r = 1)$; moreover, free paths exist from node b to node e that belong to $H_S(y, r = 1)$ with respect to node b. Another way is through node e, which belongs to $H_S(y, r + 1)$ with respect to node x. Hence, the packet is routed to the destination node.

For example, in Figure 4, (0001) and (1011) are the source and destination nodes, respectively. The total number of links (f) to route the packet from node (0001) is three. We assume that there are two busy links (m). Then, the packet is routed to one of the remaining neighboring nodes, that is (0000), which belongs to $H_S(y, r = 1)$; moreover, free paths exist from node (0000) to node (1000), which belong to $H_S(y, r = 1)$ with respect to node (0000). Another way is through node (1000), which belongs to $H_S(y, r + 1)$) with respect to node (0001). Similarly, the packet is routed in one or two steps till it reaches the destination node. In this example, the packet is routed through the given path: $(0001) \Rightarrow (0000) \Rightarrow (1000) \Rightarrow (1001) \Rightarrow (1011)$. □

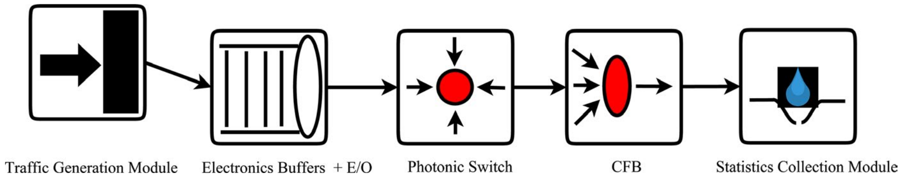

FIGURE 5 Node architecture in the simulation environment

Lemma 2 *Destination node* y *is reachable from node* x *even if* m *busy links are present at node* y *and* $r = H_D(x, y) = 1$.

Proof Let us consider a node y with m busy links such that $r \leq m < f$; therefore, the data packet d is delayed in the fiber loop for a single time slot. Hence, $f - m$ number of free links exist. The packet d will not be routed to the busy link when it attempts to reach the destination. However, it will reach the destination through one of the remaining neighboring nodes b', which belongs to $H_S(y, r+1)$; moreover, free links exist from node b' to node e', which belong to $H_S(y, r=1)$.

For example, in Figure 4, (0001) and (1011) are the source and destination nodes, respectively. The total number of links (*f*) to receive the packet from the source node is three. We assume that there are two busy links (*m*); therefore, node (0011) and node (1001) are not available. Then, the packet has to be received through one of the remaining neighboring nodes, that is (1000), which belongs to $H_S(y, r+1)$ with respect to the destination node; moreover, free paths exist from node (1000) to node (1001), which belong to $H_S(y, r=1)$ with respect to node (1011). Thus, the packet is routed in one or two steps till it reaches the destination node. □

5 | SIMULATION SETUP

5.1 | Simulation environment

The network was simulated using objective modular network testbed in C++ (OMNeT++) [29], which is a network simulation framework software. The block diagram of the single node architecture is shown in Figure 5. The entire network was built similarly. The network operated at a data rate of 100 Gbit/s. As shown in Figure 5, the traffic generation module generates the normalized load as specified in the simulation environment. Realistic DC network traffic was modeled in a similar simulation environment [30]. Each node was a 2 × 2 switch with bidirectional signaling. Each input port was connected with an ingress buffer of size equal to the packet length [8]. A controller for buffering (CFB) block was used to manage the inherent buffering.

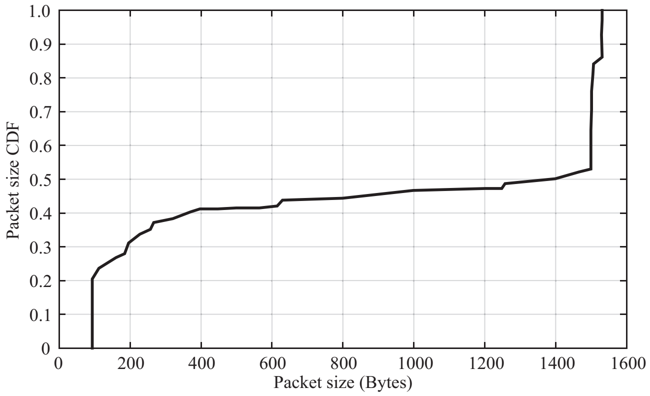

FIGURE 6 Cumulative distributed function of generated packets

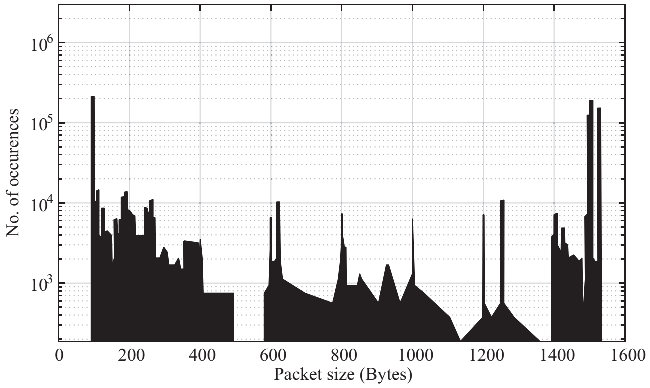

FIGURE 7 Histogram of generated packets

5.2 | Traffic generation

To analyze the traffic generation, we modeled the DC traffic based on Refs. [31–33] using the *XY* routing algorithm. The network performance was analyzed under the normalized traffic load, which varied from 0 to 1, as illustrated in Figures 8 to 13. In real time, the packet length followed a bimodal distribution; thus, the packet length varied from 40 bytes to 1500 bytes [8]. The cumulative distribution function of the packet length, as illustrated in Figure 6, depicts a bimodal distribution. The histogram of the generated packets within the simulation is shown in Figure 7. In this figure, the random occurrence of packets with variable size is visible. The results obtained using the *XY* routing algorithm were

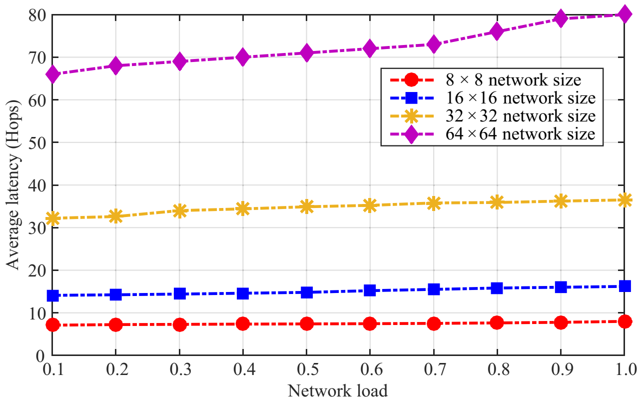

FIGURE 8 Network latency of the torus topology under the implementation of the *XY* routing algorithm for various network sizes under different network loads

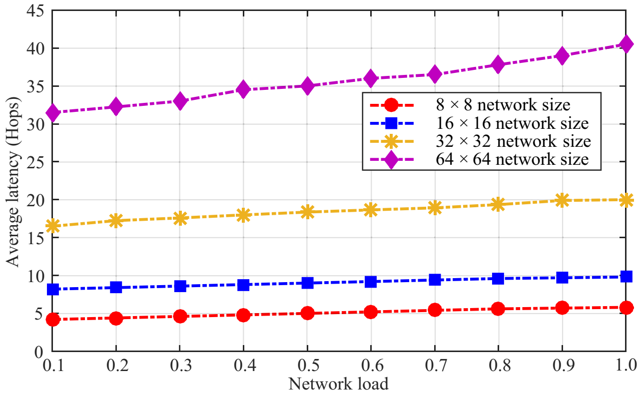

FIGURE 9 Network latency of the torus topology under the implementation of BORA for various network sizes under different network loads

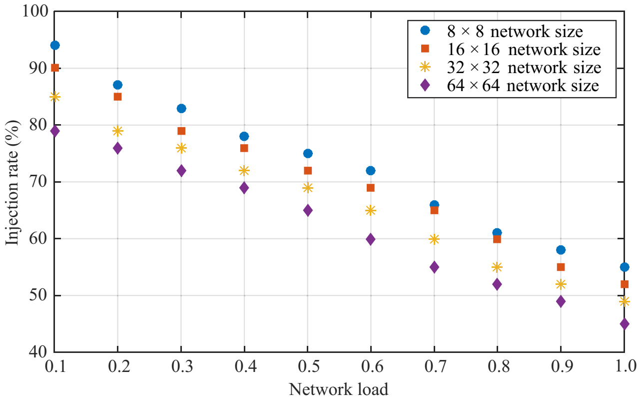

FIGURE 10 Injection rate of the torus network topology under the implementation of the *XY* routing algorithm for various network sizes under different network loads

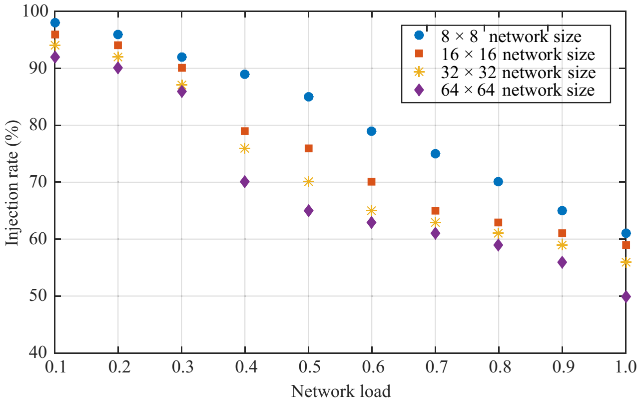

FIGURE 11 Injection rate of the torus network topology under the implementation of BORA for various network sizes under different network loads

verified with Ref. [27]. In the same simulation environment, the proposed BORA was implemented.

6 | PERFORMANCE ANALYSIS and SIMULATION

The performance of the torus network topology was analyzed and compared in terms of latency and throughput (measured in terms of injection rate and number of packets received) using the *XY* routing algorithm and BORA, as described in this section. BORA demonstrates advantages in terms of low latency and high throughput by utilizing the property of inherent buffering and by effectively utilizing the nonoverlapping paths. The performance parameters obtained by implementing the *XY* routing algorithm for network sizes of 8 × 8 and 16 × 16 were similar to that presented in [27]. Under the same conditions, we further simulated the *XY* routing algorithm for a network size of 32 × 32 and 64 × 64 using OMNeT++. BORA was also implemented using OMNeT++ [29] under the same simulation environment. The parameters used to maintain the simulation environment are listed in Table 2. The simulation was executed sufficiently to ensure the achievement of steady state.

6.1 | Network latency

This section includes the analysis and comparison of the network latency for the torus topology using the *XY* algorithm and BORA under normalized DC traffic. The network latency was analyzed in terms of the number of hops. Each packet was processed in a single time slot. In the *XY* routing algorithm, the wraparound paths are utilized to provide inherent buffering whenever the output links are not available. Otherwise, it is idle. Hence, the *XY* routing required four hops (as shown in section 2). However, the network latency (with the same source and destination nodes) using BORA was equal to two hops (as shown in section 3). Hence, there is a reduction in the network

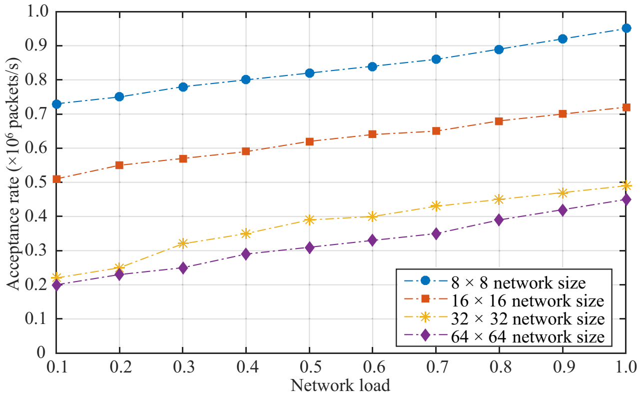

FIGURE 12 Average network throughput (measured in terms of the number of received packets) of the torus topology under the implementation of the *XY* routing algorithm for various network sizes under different network loads

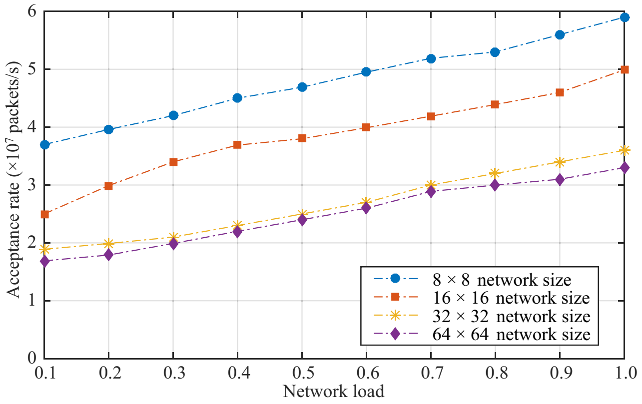

FIGURE 13 Average network throughput (measured in terms of the number of received packets) of the torus topology under the implementation of BORA for various network sizes under different network loads

latency in terms of hops by approximately 50% while using BORA. The wraparound paths in BORA are not restricted to provide inherent buffering. Whenever it is idle, it is also used for transmitting packets, without compromising the inherent buffering capacity. From the results in Figures 8 and 9, it is clear that the network latency also increases with the increase in network load. In the *XY* routing algorithm under a full load, indicated by a network size of 64 × 64, the latency was approximately 80 hops. In the case of BORA, under the same condition, the latency was approximately 40 hops only.

6.2 | Network throughput

6.2.1 | Injection rate

Injection rate is one of the network performance parameters for throughput analysis. The injection rate is defined as the total number of packets successfully injected out of the total attempts made to send the packet inside the network [1]. The injection rate of the *XY* routing protocol and BORA was compared under different traffic load conditions. The injection rate of the network decreased with the increase in network load. From Figures 10 and 11, it can be observed that the injection rate is 45% and 50% for the *XY* routing algorithm and BORA, respectively, under the full load condition indicated by a network size of 64 × 64. In the *XY* routing algorithm, the wraparound paths are used only for buffering, whereas in BORA, they are used for buffering as well as routing. Therefore, without any delay, more packets can be injected in three different paths (two routing paths and one wraparound path [whenever it is idle]) when compared to the two different paths available in the *XY* routing algorithm. Therefore, more packets can be injected in BORA; hence, the injection rate is superior in BORA.

TABLE 2 Parameters in simulation

Parameters	Value
Network size	4 × 4, 8 × 8, 16 × 16, 32 × 32, 64 × 64
Optical link latency	25.6 ns
Packet length	Variable (Range: 40 bytes to 1500 bytes)
Simulation time	1 000 000 slots
Optical link bandwidth	100 Gb/s

6.2.2 | Number of packets received

The network throughput describes the quality of the network. The relation between the network load and the number of packets received using the *XY* algorithm and BORA is shown in Figures 12 and 13, respectively. Under the full load condition, the number of packets received using BORA were 3.3×10^7 packets when compared to 0.45×10^7 packets received while using the *XY* routing algorithm for a network size of 64 × 64.

7 | CONCLUSION

The torus network topology is considered as a suitable candidate for optical interconnections because of its numerous advantages over other topologies. Traditionally, the *XY* routing algorithm is implemented in the torus network. However, in this routing algorithm, the wraparound paths that connect end nodes are used to provide only inherent buffering; otherwise, this path is idle. If these routing paths are efficiently utilized, the throughput and latency can be improved. In this paper, we proposed a novel algorithm called BORA to improve the network latency and throughput (measured in terms of injection rate and number of packets received). In this algorithm, without compromising the inherent buffering property, the routes were optimized and performance was improved. The shortest nonoverlapping paths with high priority were calculated by the hamming shell and used for routing purposes. Therefore, the wraparound paths were efficiently used to route the packet whenever they were idle. This in turn decreased the latency by 50% and increased the throughput in terms of both the injection rate and the number of packets received.

ORCID

Sangeetha Rengachary Gopalan https://orcid.org/0000-0001-9818-8934

REFERENCES

1. A. Sharma and R. G. Sangeetha, *Comparative study of optical interconnection architectures in data center networks*, J. Opt. Commun. **40** (2019), 225–238.
2. S. L. Gaixas et al., *Scalable topological forwarding and routing policies in RINA-enabled programmable data centers*, Trans. Emerg. Telecommun. Technol. **28** (2017), no. 12, e3256.
3. M. Jahanshahi and F. Bistouni, *Improving the reliability of the Benes network for use in large-scale systems*, Microelectron. Reliab. **55** (2015), 679–695.
4. A. Birolini, *Reliability Engineering—Theory and Practice*, (5th ed.), Springer, Berlin, 2007.
5. Cisco Visual Networking, *Cisco global cloud index: Forecast and methodology, 2012–2017*, white paper, 2013.
6. N. Farrington et al., *Helios: a hybrid electrical/optical switch architecture for modular data centers, ACM SIGCOMM*, Comput. Commun. Rev. **41** (2011), no. 4, 339–350.
7. P. Costa et al., *Camcube: a key-based data center*, Technical Report MSR TR-2010-74 Microsoft Research (2010)
8. N. Calabretta et al., *On the performance of a large-scale optical packet switch under realistic data center traffic*, J. Opt. Commun. Netw. **5** (2013), no. 6, 565–573.
9. M. Keri et al., *Performance of a routing and wavelength assignment scheme for an optical-carrier-reusable ring network*, IET Netw. **3** (2014), no. 4, 267–274.
10. C. Hawkins et al., *The data vortex, an all optical path multicomputer inter-connection network*, IEEE Trans. Parall. Distr. Syst. **18** (2007), no. 3, 409–420.
11. C. Raffaelli, *Design of a multistage optical packet switch*, Eur. Trans. Telecommun. **11** (2000), no. 5, 443–451.
12. A. Shacham and K. Bergman, *Optimizing the performance of a data vortex interconnection network*, J. Opt. Netw. **6** (2007), 369–374.
13. J. Xiao et al., *Scalable data center network architecture with distributed placement of optical switches and racks*, IEEE/OSA J. Opt. Commun. Netw. **6** (2014), 270–281.
14. A. Sharma and R. G. Sangeetha, *Reliability analysis of data center network*, Springer, Optical And Microwave Technologies, 2018, pp. 71–80.
15. R. G. Sangeetha, V. Chandra, and D. Chadha, *Optical interconnection bidirectional data vortex network: Architecture and performance analysis*, J. Lightwave Technol. **31** (2013), no. 8, 1283–1294.
16. V. Chandra, D. Chadha, and R. G. Sangeetha, *Bidirectional optical data packet switching interconnection network*, U.S. Patent No. 9,031,407. 12 May 2015.
17. S. Mishra et al., *Hardware implementation of optical switching node for data center networks*, Microw. Opt. Technol. Lett. **61** (2019), no. 3, 843–846.
18. V. Yadav et al., *Hardware implementation of contention aware optical switching node for data center networks*, Microw. Opt. Techn. Let.. **61** (2019), no. 10, 2434–2440.
19. K. –I. Kitayama et al., *Torus-topology data center network based on optical packet/agile circuit switching with intelligent flow management*, J. Lightwave Technol. **33** (2015), 1063–1071.
20. A. Sharma and R. G. Sangeetha, *Performance analysis of high speed low-latency torus optical network*, in Proc. Int. Conf. Commun. Syst. Netw. (Bengaluru, India), 2018, pp. 488–491.
21. G. P Joseph and M. Syska, *Circuit-switched broadcasting in torus networks*, IEEE Trans. Parall. Distrib. Syst. **7** (1996), no. 3, 246–255.
22. D. F. Robinson, P. K. McKinley, and B. H. C. Cheng, *Optimal multicast communication in wormhole-routed torus networks*, IEEE Trans. Parall. Distrib. Syst. **6** (1995), no. 10, 1029–1042.
23. Y. J. Tsai and P. K. McKinley, *A broadcast algorithm for all-port wormhole-routed torus networks*, IEEE Trans. Parall. Distrib. Syst. **7** (1996), no. 8, 876–885.
24. F. Yan et al., *Opsquare: A flat dcn architecture based on flow-controlled optical packet switches*, IEEE/OSA J. Opt. Commun. Netw. **9** (2017), 291–303.
25. W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*, Elsevier, 2004.
26. R. Takahashi et al., *Torus data center network with smart flow control enabled by hybrid optoelectronic routers*, IEEE/OSA J. Opt. Commun. Netw. **7** (2015), no. 12, B141–B152.
27. X. Qi et al., *Boin: A novel bufferless optical interconnection network for high performance computer*, in Proc. IEEE/ACS Int. Conf. Comput. Syst. Applicat. (Rabat, Morocco), May 2009, pp. 117–123.
28. Q. Xingyun et al., *A fault tolerant bufferless optical interconnection network*, in Proc. IEEE/ACIS Int. Conf. Comput. Inf. Sci. (Shanghai, China), June 2009, pp. 249–254.
29. A. Varga, Omnet++ community site, http://www.omnetpp,org/ (2007).
30. A. Shakeri et al., *Traffic allocation strategies in WSS-based dynamic optical networks*, J. Opt. Commun. Netw. **9** (2017), no. 4, B112–B123.
31. T. Benson et al., *Understanding data center traffic characteristics*, in Proc. ACM Workshop Res. Enterprise Netw. (Barcelona Spain), Aug. 2009, pp. 65–72.
32. T. Benson, A. Akella, and D. A. Maltz, *Network traffic characteristics of data centers in the wild*, in Proc. ACM SIGCOMM Conf. Internet Meas. (Melbourne, Australia), Nov. 2010, pp. 267–280.
33. S. Kandula et al., *The nature of data center traffic: Measurements & analysis*, in Proc. ACM SIGCOMM Conf. Internet Meas. (Chicago, Illinois, USA), Nov. 2009, pp. 202–208.

AUTHOR BIOGRAPHIES

Abhilasha Sharma received her BTech degree in Electronics and Communication Engineering in 2011 from Rajasthan Technical University, Kota. She received her MTech degree in Digital Communication in 2014 from Rajasthan Technical University, Kota. She is currently pursuing her PhD at VIT University, Chennai. Her areas of interest are data center optical networks, optical fiber communication, fault tolerance studies, and free space optical communication.

Sangeetha Rengachary Gopalan's education includes a Bachelor of Engineering in Electronics and Communication from Bharathidasan University, Trichy. She completed her master's degree in Computer and Communication Engineering from Anna University, Chennai. Her doctoral studies were conducted at Indian Institute of Technology, Delhi, where she completed her PhD in Optical Networks. Her primary research interests are fiber optic communications, optical networks, and free space communication. Recently, she completed a project titled "Test Bed for Hardware Implementation of All Optical Bi-Directional Switching Node" under the Fast Track Young Scientist Scheme, funded by the Department of Science and Technology, Government of India. She is currently working as an Associate Professor in SENSE, VIT University, Chennai. She is a senior member of IEEE and OSA societies.