



A scalable AWG-based data center network for cloud computing



Gang Wu^{a,c}, Huaxi Gu^{a,*}, Kun Wang^b, Xiaoshan Yu^a, Yantao Guo^c

^a State Key Laboratory of ISN, Xidian University, 710071 Xi'an, China

^b School of Computer Science and Technology, Xidian University, 710071 Xi'an, China

^c Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, 050000 Shi Jiazhuang, China

ARTICLE INFO

Article history:

Received 9 June 2014

Received in revised form

16 October 2014

Accepted 6 December 2014

Available online 16 December 2014

Keywords:

Data center networks
Arrayed Waveguide Grating
Optical interconnection
Multi-path routing

ABSTRACT

With the development of cloud computing and other online applications, the traffic for data center network (DCN) has increased significantly. Therefore, it is extremely important for DCNs to support more and more servers and provide high scalability, high throughput and low latency. Some current topologies for data centers have such inherent problems as poor scalability, lack of path diversity, cabling complexity, etc. This paper proposes a scalable AWG-based optical interconnection network for data centers, which is called OIT. OIT possesses good scalability and path diversity and benefits from the inherent parallelism and high capacity of WDM and AWG, which makes it a suitable candidate topology for data centers in the cloud computing era. A multi-path routing algorithm is also designed to utilize OIT's parallel links and distribute the load more evenly. The simulation results show that the packet latency and network throughput performance of OIT is better than that of fat tree topology under uniform random distribution or 50%, 80% intra pod traffic distribution and different packet sizes.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Over the recent years, data centers are facing an exponential increase of the network traffic due to the rise of cloud computing and other emerging online applications. Many of these applications are data-intensive and require high interaction between servers, which imposes greater pressure on the interconnection and communication schemes of the data center [1]. However, traditional DCNs based on tree shaped topology can hardly meet such requirements as the traffic aggregate in the top of the tree and the root switch becomes the bottleneck. Also, its scalability is severely limited by the performance of the root switch.

Some new topologies based on electrical switching have been proposed. Fat tree [2] is a pod based topology which can deliver large bisection bandwidth and has widely been adopted, but it faces the problems of limited scalability and downlink's inflexibility. DCell [3], BCube [4] and MDCube [5] are recently proposed network architectures for modular data centers. DCell is a recursively defined, high network capacity structure with mini-switches to interconnect servers. But due to its structural features, DCell has some inherent defects. The irregular network topology makes it difficult to deploy a cabling solution and the network traffic in DCell is nonuniformly distributed (most traffic is concentrated in the lower level). BCube is another server centric network structure that is built with multiple network ports. However, BCube adopts too many mini-switches, thus makes it difficult for enterprises to build large data centers, and deploying a cabling solution is also

* Corresponding author. Tel.: +86 13359252600.
E-mail address: hxgu@xidian.edu.cn (H. Gu).

complex in BCube. MDCube, built recursively with BCube containers, deploys optical fibers to interconnect multiple BCube containers by using the high speed interfaces of COTS switches. But MDCube also has some defects such as large network diameter and complex cabling solution.

To mitigate the effects brought by huge traffic and meet the requirements of cloud computing era, optical interconnects emerged as promising solutions that can provide high bandwidth with reduced power consumption [6,7]. Some schemes are based on Micro-Electro-Mechanical Systems Switches (MEMS switches) such as c-through [8], Helios [9,10], Proteus [11] and OSA [12]. However the reconfiguration of the MEMS switch requires several milliseconds, thus these themes are not appropriate for delay sensitive applications in the cloud computing environment. Another class of schemes is based on an all-optical switching fabric called Arrayed Waveguide Grating (AWG) that has been proven in telecom applications to scale to petabit/second aggregate switching capacity [13–15], such as DOS [16] and LIONS [17]. Nevertheless, AWG suffers from deviation of passband center frequencies and the crosstalk, both of which becomes very large as the port number of AWG increases [18,19]. Considering the factor of the deviation and crosstalk, the current port count of AWG could only reach about 128 [19], thus the scalability of these schemes is severely limited. Some researchers have proposed solutions such as cascading small AWGs to form a large scale switch [19,20], or applying AWG to Clos topology to settle the problem [21,22]. But the complexity and redundant cost of DCN network construction increases significantly as the network size increases.

In this paper, we propose a scalable AWG-based optical network named as OIT (optical interconnect topology). OIT is a cluster based topology which adopts low-radix AWGs and ToR switches to form clusters, and then these clusters are interconnected by multiple WDM fibers to construct a

large network. By adopting small AWGs, OIT can still easily scale out to hundreds of thousands of servers. The structural features ensure OIT with good path diversity. A multi-path routing algorithm is also designed according to OIT’s multi-path characteristic. Theoretical analyses show that OIT achieves good scalability while keeps network diameter at a constant low value. Simulation results demonstrate that OIT has much saturation bandwidth improvement over fat tree topology under different traffic distributions and packet sizes.

The rest of the paper is organized as follows. Section 2 gives a full description of the addressing, interconnection, scaling pattern and path diversity of OIT. Section 3 provides the communication mechanism of OIT. Section 4 presents the theoretical analysis and performance evaluation of OIT. Finally, Section 5 briefly concludes the paper.

2. The topology of OIT

2.1. The interconnection rules

Fig. 1 gives an overview of OIT(N, m) structure. It composes of $N+1$ clusters and each cluster consists of two layers of AWGs and one layer of server racks. Specially, in one cluster there are $2 \times N$ AWGs and $N \times N$ server racks ($N \geq 2$), and a server rack is made up of a ToR switch and m servers. WDM fibers are adopted to connect different clusters and devices in each cluster. We denote each layer-1 server rack with a 3-tuple $[clusterid, layerid, rackid]$. The $clusterid$ defines the cluster number and takes values from 1 to $N+1$. The $layerid$ is defined as 1. The $rackid$ represents the server rack number and takes value from 1 to $N \times N$ from left to right. Then we mark each layer-2 AWG with a 3-tuple $[clusterid, layerid, awgid]$. The $layerid$ is defined as 2 and $awgid$ takes value from 1 to N from left to right. Similarly, each layer-3 AWG is also

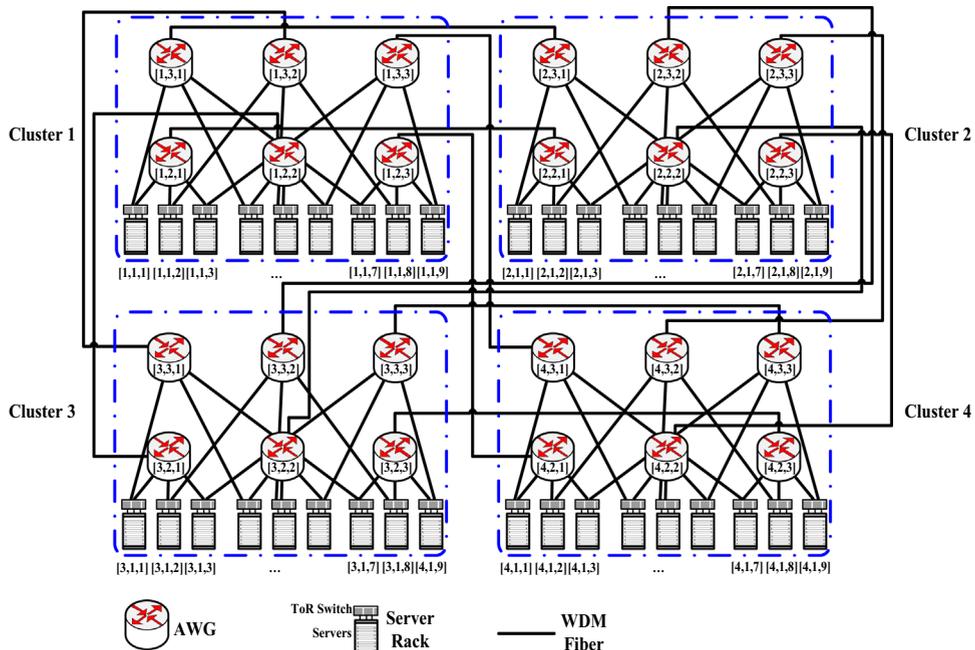


Fig. 1. The OIT($N=3$) topology.

assigned a 3-tuple [*clusterid*, *layerid*, *awgid*] and the *layerid* is 3.

Connection Rule:

- 1) Connection within a cluster: Each layer-2 AWG connects server racks whose *rackid* ranges from $(awgid-1) \times N+1$ to $awgid \times (N+1)$. Each layer-3 AWG connects server racks whose *rackid* divides N remains the same as *awgid* (if the remainder is 0, we take it as N).
- 2) Connection between clusters: The last port of AWG is used to connect to another AWG in a different cluster. Let [*clusterid*, *awgid*] denote each AWG. Then two AWGs in different clusters of the same *layerid* and denoted with 2-tuples [*clusterid*, *awgid*-1] and [*awgid*, *clusterid*] are connected by a link for every *clusterid* and every $awgid > clusterid$.

Take the OIT($N=3$) in Fig. 1 as an example, which adopts 4-port AWGs to build a 4-cluster structure. Inside cluster 1, layer-2 AWG (*awgid*=1) connects to server racks whose *rackid* equals from 1 to 3. Layer-3 AWG (*awgid*=1) connects to server racks whose *rackid* equals 1, 4 and 7. Other AWGs and server racks are interconnected in a similar way. In the case of connection between different clusters, layer-2 and layer-3 AWG (*awgid*=1) in cluster 1 are used to connect to the AWGs (*awgid*=1) in cluster 2 respectively. Similarly the AWGs (*awgid*=2 and *awgid*=3) in cluster 1 are connected to the AWGs (*awgid*=1) in cluster 3 and cluster 4 respectively. By this analogy, AWGs and server racks are interconnected according to the connection rule.

2.2. The structure of the ToR switch and the routing characteristic of AWG

The structure of the ToR switch is shown in Fig. 2. It can be noticed that each ToR switch adopts two ports to connect a layer-2 AWG and a layer-3 AWG respectively. Port 1 is connected to layer-2 AWG and port 2 to layer-3 AWG, and the rest ports are used to connect the servers. The routing calculation of OIT is performed in the routing control module. $N+1$ wavelengths are used in our topology, thus each port of the ToR switch which connected to AWG maintains $N+1$ queues, $N+1$ O/E (E/O) converters, an optical DEMUX and MUX.

AWGs are passive data-rate independent optical devices that route each wavelength of an input to a different output. The cyclic wavelength routing characteristic of the AWG allows different inputs to reach the same output simultaneously by using different wavelengths. The sequence number of the wavelength s_w can be calculated by formula (1)

$$s_w = (p_o + p_i - 1) \bmod n_w \quad (1)$$

p_o and p_i represent the output and input port number respectively, and n_w is the total number of wavelength used in the network. Formula (2) can be derived from formula (1)

$$p_o = (s_w - p_i + n_w) \bmod n_w + 1 \quad (2)$$

Table 1 demonstrates an example of the wavelength connection pattern for an 8×8 AWG. The cyclic wavelength routing characteristic of the AWG can be clearly found.

2.3. The scaling pattern

We use n_t to represent the total number of the server racks in a OIT(N). An OIT(N , m) contains $N+1$ clusters and each cluster consists of N^2 server racks. Thus we can derive formula (3)

$$n_t = (N+1) \times N^2 \quad (3)$$

With the number N increasing, the number of the server racks in a cluster increases and the number of clusters increases. Therefore, an OIT with a small number N can support thousands of server racks. Considering the fact of the crosstalk and the deviation, the port number of AWG can reach 128. In OIT each AWG adopts one port to connect to the AWG in another cluster and has N free ports for the ToR switches. A typical ToR switch can support as many as twenty servers, and in OIT two ports of the ToR switch are reserved for connection with AWG, thus the number of the servers a rack contains (m) can take value from 1 to 18 and be adjusted flexibly according to actual demand.. If the maximum port number of AWG is set to 100, then N can reach 99. According to formula (3), our proposal can support as many as 980k server racks (980k to 17.6 million servers), which is suitable for current data centers.

3. Communication and routing process

From its topology, we can notice that OIT has multiple paths between different server racks in the same cluster and between different clusters. Thus a multi-path routing algorithm is designed to utilize OIT's parallel links and distribute the load more evenly. If there is more than one path from the source server rack to the destination rack, the packet will be randomly delivered on one of the suitable paths. We use "src" to represent the source server rack and "dest" to represent the destination server rack. For each ToR switch, the port number is defined as 1 when the port is connected to a layer-2 AWG and 2 when connected to a layer-3 AWG. For each layer-2 AWG, the port number is the same as the *rackid* of the server rack that the port connects to, and the port number is defined as $N+1$ if the port is used to connect to an AWG in another cluster. For each layer-3 AWG, the port number increases from 1 to N as the *rackid* of the server rack that it connects to increases, and the port number is defined as $N+1$ if the port is used to connect to an AWG in another cluster.

When the ToR switch receives a packet (whether from servers or AWGs), the routing control module will judge the packet's destination server rack address. If the destination server is in the same rack of the local ToR switch, the packet will be immediately forwarded to the destination server, otherwise the routing control module will calculate the wavelength needed to forward the packet to the correct output port in the next hop AWG and insert the packet into the corresponding queue of port 1 or port 2. In the example of Fig. 1, if server rack [1,1,1] needs to communicate to server rack [2,1,8], one of the two

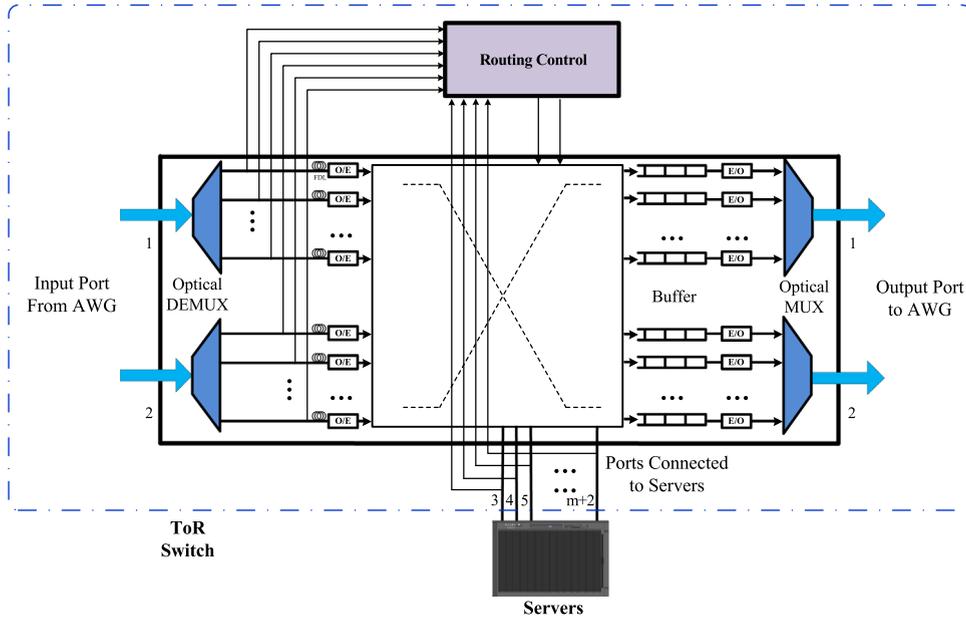


Fig. 2. The structure of the ToR switch in OIT.

Table 1
Wavelength connection pattern for an 8 × 8 AWG.

	Input ports p_i								
	1	2	3	4	5	6	7	8	
Output ports p_o	1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	1	
3	3	4	5	6	7	8	1	2	
4	4	5	6	7	8	1	2	3	
5	5	6	7	8	1	2	3	4	
6	6	7	8	1	2	3	4	5	
7	7	8	1	2	3	4	5	6	
8	8	1	2	3	4	5	6	7	

available paths $[1,1,1] \rightarrow [1,2,1] \rightarrow [2,2,1] \rightarrow [2,1,2] \rightarrow [2,3,2] \rightarrow [2,1,8]$ or $[1,1,1] \rightarrow [1,3,1] \rightarrow [2,3,1] \rightarrow [2,1,7] \rightarrow [2,2,3] \rightarrow [2,1,8]$ will be chosen randomly. The pseudocode of routing algorithm is shown below. We use $[D_1, D_2, D_3]$ to represent the 3-tuple address of the destination server rack, and $[L_1, L_2, L_3]$ to represent the 3-tuple address of the local current node.

Routing algorithm of OIT

Input: destination address $[D_1, D_2, D_3]$,
current local node address $[L_1, L_2, L_3]$

Output: output port

```

01 if ( $L_2=1$ )/the local node is a ToR switch*/
02 {
03     if ( $D_1=L_1$ )/dest is in the same cluster*/
04     if ( $D_2=L_2$ )/dest is connected to the same layer-2
    AWG*/
05         if ( $D_3=L_3$ )/dest is in the local current rack*/
06         end;
07     else/dest is connected to the same layer-2
    AWG*/
08         output port=1;

```

```

09     else/dest is not connected to the same layer-2
    AWG */
10         output port=2;
11     else/dest is not in the same cluster*/
12     {
13         if ( $((L_1-1)=D_3 \ \&\& \ L_3=D_1) \ \parallel \ ((D_1-1)=L_3$ 
14              $\ \&\& \ D_3=L_1)$ )
15             /*layer-2 AWG has a link to the dest cluster*/
16             output port=1;
17         else
18             output port=rand()%2+1;
19     }
20 }
21 if ( $L_2=2$ )/the local node is a layer-2 AWG*/
22 {
23     if ( $D_1=L_1$ )/dest is in the same cluster */
24     if ( $D_3=L_3$ )/dest is connected to the local layer-2
    AWG*/
25         output port= $D_3$ ;
26     else/dest is not connected to the local layer-2 AWG
    */
27         output port=rand()%N+1;
28     else/dest is not in the same cluster */
29     {
30         if ( $((L_1-1)=D_3 \ \&\& \ L_3=D_1) \ \parallel \ ((D_1-1)=L_3$ 
31              $\ \&\& \ D_3=L_1)$ )
32             /*local layer-2 AWG has a link to the dest
    cluster*/
33             output port= $N+1$ ;
34     else/dest will be sent to a server whose layer-3
    AWG has a link to the dest cluster */
35     {
36     }
37     if  $D_1 > 1$ 
38         output port= $D_1$ ;
39     else
40         output port=1;
41 }
42 }
43 }
44 if ( $L_2=3$ )/the local node is a layer-3 AWG*/

```

```

45 {
46   if (D1==L1)/*dest is in the same cluster*/
47     output port=D3;
48   else/*dest is not in the same cluster*/
49     {
50       if (((L1-1)==D3 && L3==D1)||((D1-1)==L3
51         && D3==L1))
52         /*local layer-3 AWG has a link to the dest
cluster*/
53         output port=N;
54       else/*the packet will be sent to a server whose
layer-2
55         AWG has a link to the dest cluster*/
56         {
57           if D1 > 0
58             output port=D1;
59           else
60             output port=1;
61         }
62     }
63 }

```

4. Evaluation and analysis

Fat tree is a pod based topology which has been extensively studied and widely adopted in current data center networks [23–25]. BCube and DCell are high scalable structures which adopt mini-switches and multiport servers to build data centers in a recursive way. As is shown in Table 2, we compare OIT with these topologies in basic network parameters such as network size, network diameter, bisection bandwidth and node degree. In Table 2, “*b*” represents the number of servers one switch in BCube has. “*k*” represents the number of pods in fat tree. “*d*” represents the number of the layers in DCell or BCube. “*t*” represents the total servers the network can support.

Network diameter is an important parameter in determining the average latency of a topology, while the maximum number of servers a topology can support is significant to judge its scalability. OIT can support a lot more servers than fat tree while keeps its network diameter at 7, just 1 more than fat tree’s network diameter. BCube is a server centric network structure designed for modular data centers. Although BCube has good scalability, the network diameter of BCube increases with a multiplication factor of 2. Besides, BCube adopts too many mini-switches, thus makes it difficult for enterprises to build large data centers, and deploying a cabling solution is also complex in BCube. Above all, taking both network diameter and scalability into consideration, OIT achieves a better tradeoff between these two parameters.

Fat tree is a pod based topology which is widely adopted in current data center networks. Like fat tree, OIT is also a cluster based topology, and they both use three layers of switches to

Table 2
Basic Parameters Of OIT And Other Topologies.

	Number of servers	Network diameter	Bisection width	Node degree
Fat tree	$k^3/4$	6	$t/2$	
BCube	b^{d+1}	$2(d+1)$	–	$d+1$
DCell	–	$2^{d+1}-1$	$t/(4 \log_2 t)$	$d+1$
OIT	$m \times (N+1) \times N^2$	7	$m \times (N+1)^2/2$	$N+1$

interconnect servers (racks). Since they belong to the same type of architecture, we have built simulation platforms for fat tree and OIT(3, 5) of the similar network size based on OPNET software. Extensive simulations are conducted to evaluate the performance of the proposed OIT network. The simulation environment is set as follows: customized credit-based flow control mechanism is employed. The bandwidth of electrical link is set to 15 Gbps and each WDM fiber carries four 15 Gbps wavelengths. Based on the analysis of the traffic characteristic in [26], the packet size is set to 512 bytes and 768 bytes. Each server generates packets continuously, and the interval time between two packets generated on one server follows a negative exponential distribution.

From available data center network measurement studies [26,27], we can find that the workloads of data centers exhibit a high degree of variance. According to [28], we note that nearly 75% on average of server-generated traffic in the cloud data centers of the example is confined to within the rack in which it was generated, and in university and private enterprise data centers of the example, at least 50% of the server-originated traffic leaves the racks.

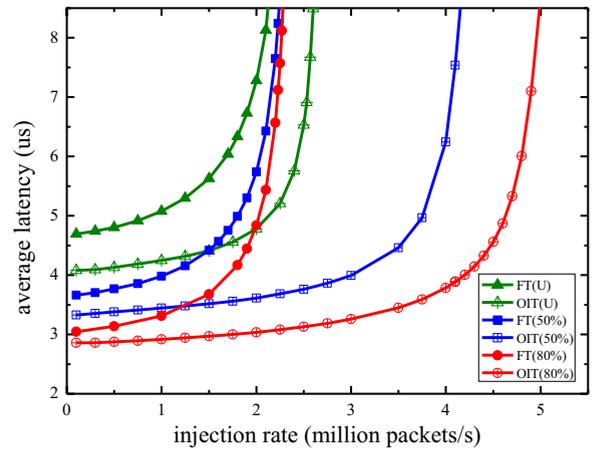


Fig. 3. The average latency vs injection rate of fat tree and OIT with packet size of 512 bytes.

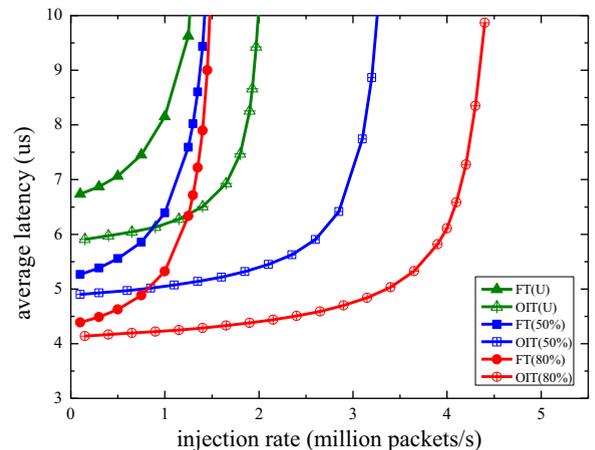


Fig. 4. The average latency vs injection rate of fat tree and OIT with packet size of 768 bytes.

Figs. 3 and 4 plots the average packet latency vs injection rate of fat tree and OIT under different traffic distributions with packet sizes of 512 bytes and 768 bytes respectively. In these Figs, “FT” stands for “fat tree”, “U” stands for “uniform traffic pattern”, which means the destination address of each packet is generated randomly. “50%” and “80%” means the destination address of each packet has a probability of 50% or 80% to stay in the same pod (cluster) with the source server.

It can be seen that the latency performance of OIT is much better than that of fat tree under uniform traffic distribution or 50%, 80% intra pod (cluster) traffic distribution. As more fraction of total traffic is confined to the same cluster, the better latency performance OIT can achieve, which proves that OIT has the good ability to support such applications with strong locality. The main reason that OIT can achieve lower packet latency than fat tree is because the wavelengths have been assigned at the ToR switch, so packets can be directly forwarded as they enter AWGs and multiple packets can be transmitted in AWGs simultaneously due to the wavelength routing characteristic. However, in fat tree packets have to go through queuing and routing computing process each time they enter a switch, which great increases the packet latency. Based on our analysis, OIT can achieve lower packet latency than fat tree. Moreover, the multi-path routing algorithm of OIT utilizes the parallel paths in the structure while fat tree's routing algorithm cannot make full use of its path diversity, and this also contributes to OIT's better network performance.

5. Conclusion

In this paper, we propose a scalable AWG-based optical interconnection network for data centers, which is called OIT. A multi-path routing algorithm is also designed according to its network features. In our theoretical analysis, OIT has good basic network properties such as network diameter and high scalability. The simulation results show that the latency performance of OIT outperforms fat tree topology and can better deal with traffic with strong locality, which indicates that OIT is a good candidate for scalable optical data center networks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China Grant no. 61472300 and 61334003, the Fundamental Research Funds for the Central Universities Grant no. JB142001-5, the 111 Project Grant no. B08038.

References

- [1] X. Zhao, V. Vusirikala, B. Koley, et al., The prospect of inter-data-center optical networks, *IEEE Commun. Mag.* 51 (2013) 9.
- [2] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, *ACM SIGCOMM Comput. Commun. Rev.* 38 (4) (2008) 63–74.
- [3] C. Guo, H. Wu, K. Tan, et al., DCell: a scalable and fault-tolerant network structure for data centers, *ACM SIGCOMM Comput. Commun. Rev.* 38 (4) (2008) 75–86.
- [4] C. Guo, G. Lu, D. Li, et al., BCube: a high performance, server-centric network architecture for modular data centers, *ACM SIGCOMM Comput. Commun. Rev.* 39 (4) (2009) 63–74.
- [5] H. Wu, G. Lu, D. Li, et al., MDCube: a high performance network structure for modular data center interconnection, in: *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. ACM, 2009, pp. 25–36.
- [6] A. Jajszczyk, Optical networks—the electro-optic reality, *Opt. Switch. Netw.* 1 (1) (2005) 3–18.
- [7] C. Kachris, I. Tomkos, A survey on optical interconnects for data centers, *IEEE Commun. Surv. Tutor.* 14 (4) (2012) 1021–1036.
- [8] G. Wang, D.G. Andersen, M. Kaminsky, et al., c-Through: part-time optics in data centers, *ACM SIGCOMM Comput. Commun. Rev.* 40 (4) (2010) 327–338.
- [9] N. Farrington, G. Porter, S. Radhakrishnan, et al., Helios: a hybrid electrical/optical switch architecture for modular data centers, *ACM SIGCOMM Comput. Commun. Rev.* 41 (4) (2011) 339–350.
- [10] N. Farrington, Y. Fainman, H. Liu, et al., Hardware requirements for optical circuit switched data center networks, in: *Proceedings of the Optical Fiber Conference (OFC/NFOEC'11)*, 2011.
- [11] A. Singla, A. Singh, K. Ramachandran, et al., Proteus: a topology malleable data center network, in: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ACM, 2010, p. 8.
- [12] K. Chen, A. Singla, A. Singh, et al., OSA: an optical switching architecture for data center networks with unprecedented flexibility, *IEEE/ACM Transactions on Networking* 22 (2) (2014) 498–511.
- [13] K.A. McGreer, Arrayed waveguide gratings for wavelength routing, *IEEE Commun. Mag.* 36 (12) (1998) 62–68.
- [14] S.J.B. Yoo, Optical packet and burst switching technologies for the future photonic internet, *J. Lightwave Technol.* 24 (12) (2006) 4468–4492.
- [15] S.J.B. Yoo, H.J. Lee, Z. Pan, et al., Rapidly switching all-optical packet routing system with optical-label swapping incorporating tunable wavelength conversion and a uniform-loss cyclic frequency AWGR, *IEEE Photon. Technol. Lett.* 14 (8) (2002) 1211–1213.
- [16] X. Ye, Y. Yin, S.J.B. Yoo, et al., DOS: a scalable optical switch for datacenters, in: *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ACM, 2010, p. 24.
- [17] Y. Yin, R. Proietti, X. Ye, et al., LIONS: an AWGR-based low-latency optical switch for high-performance computing and data centers, *IEEE J. Sel. Top. Quantum Electron.* 19 (2) (2013) 3600409.
- [18] Y.D. Jin, Q. Jiang, M. Kavehrad, Performance degradation due to crosstalk in multiwavelength optical networks using dynamic wavelength routing, *IEEE Photon. Technol. Lett.* 7 (10) (1995) 1210–1212.
- [19] K. Sato, H. Hasegawa, T. Niwa, et al., A large-scale wavelength routing optical switch for data center networks, *IEEE Commun. Mag.* 51 (9) (2013) 46–52.
- [20] T. Niwa, H. Hasegawa, K.I. Sato, et al., Large port count wavelength routing optical switch consisting of cascaded small-size cyclic arrayed waveguide gratings, *IEEE Photon. Technol. Lett.* 24 (22) (2012) 2027–2030.
- [21] H.J. Chao, K.L. Deng, Z. Jing, Petastar: a petabit photonic packet switch, *IEEE J. Sel. Areas Commun.* 21 (7) (2003) 1096–1112.
- [22] R. Proietti, Z. Cao, Y. Li, S.J.B. Yoo, Scalable and distributed optical interconnect architecture based on AWGR for HPC and data centers, in: *Proceedings of the Optical Fiber Communication Conference (OFC)*, Paper Th2A.59, 2014.
- [23] R. Niranjana Mysore, A. Pamboris, N. Farrington, et al., Portland: a scalable fault-tolerant layer 2 data center network fabric, *ACM SIGCOMM Comput. Commun. Rev.* 39, , 2009, 39–50.
- [24] Z. Guo, Y. Yang, On nonblocking multicast fat-tree data center networks with server redundancy, *IEEE Trans. Comput.* (2014) 1.
- [25] Z. Guo, J. Duan, Y. Yang, On-line multicast scheduling with bounded congestion in fat-tree data center networks, *IEEE J. Sel. Areas Commun.* 32 (1) (2014) 102–115.
- [26] T. Benson, A. Akella, D.A. Maltz, Network traffic characteristics of data centers in the wild, in: *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ACM, 2010, pp. 267–280.
- [27] T. Benson, A. Anand, A. Akella, et al., Understanding data center traffic characteristics, *ACM SIGCOMM Comput. Commun. Rev.* 40 (1) (2010) 92–99.
- [28] S. Kandula, S. Sengupta, A. Greenberg, et al., The nature of data center traffic: measurements & analysis, in: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ACM, 2009, pp. 202–208.