IEICE Electronics Express, Vol.9, No.3, 172–178

# A virtual hierarchical optical mesh based data center network

**Peibo Xie[1], Huaxi Gu[1a)], Yuan Liu[2], Xiangbin Wu[2], Hua You[3], and Dong Liu[2]**

[1] *State Key Laboratory of ISN, Xidian University, Xi'an, China*

[2] *Intel Labs China, Beijing, China*

[3] *Intel Asia-Pacific R&D Ltd, Shanghai, China*

a) *hxgu@xidian.edu.cn*

**Abstract:** Reducing the routing table size to obtain better scalability of network is more and more important for data center networks in cloud computing era. In this paper, virtual hierarchical mesh networks with different agent selection methods of different subnets are proposed to solve the network scaling embarrassment problem. Theoretical analysis and simulation results show the multi-agent selection methods outperform the single one in terms of end-to-end latency and load balance.

**Keywords:** data center, virtual hierarchical mesh, E/O-O/E conversions, deadlock avoidance, scalability

**Classification:** Fiber optics, Microwave photonics, Optical interconnection, Photonic signal processing, Photonic integration and systems

## References

[1] M. Al-Fares, A. Loukissas, and A. Vahdatet, "A scalable, commodity data center network architecture," *Proc. ACM SIGCOMM 2008 Conf. Data communication*, Seattle, WA, USA, ACM, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[2] A. Greenberg, R. H. James, and N. Jain, "VL2: a scalable and flexible data center network," *Proc. ACM SIGCOMM 2009 Conf. Data communication*, vol. 39, no. 4, pp. 51–62, Aug. 2009.

[3] G. Chuanxiong, W. Haitao, and T. Kun, "Dcell: a scalable and fault-tolerant network structure for data centers," *Proc. ACM SIGCOMM 2008 Conf. Data communication*, Seattle, WA, USA, ACM, vol. 38, no. 4, pp. 75–86, Aug. 2008.

[4] G. Chuanxiong, L. Guohan, and L. Dan, "BCube: a high performance, server-centric network architecture for modular data centers," *Proc. ACM SIGCOMM 2009 Conf. Data communication*, Barcelona, Spain, ACM, vol. 39, no. 4, pp. 63–74, Aug. 2009.

[5] Hussam, L. Abu and C. Paolo, "Symbiotic routing in future data centers," *Proc. ACM SIGCOMM 2010 Conf. SIGCOMM*, New Delhi, India, ACM, vol. 40, no. 4, pp. 51–62, Aug.-Sept. 2010.

[6] P. Lucian, R. Sylvia, and L. Gianluca, "A cost comparison of datacenter network architectures," *Proc. 6th International Conf. CoNEXT 2010*, Philadelphia, Pennsylvania, ACM, pp. 1–12, Dec. 2010.



DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

[7] M. Glick, "Optical Interconnects in Next Generation Data Centers: An End to End View," *Proc. 16th IEEE Symposium on High Performance Interconnects*, pp. 178–181, Aug. 2008.
[8] L. D. Chares and A. F. Benner, "Optics in Future Data Center Networks," *Proc. 2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI)*, pp. 104–108, Aug. 2010.

## 1 Introduction

Data center networks, which serve as the fundamental infrastructure of cloud computing, play undoubtedly important roles in distributing the processors, memory, network bandwidth and storage. Traditional data center networks employ tree structure and expensive high-end switches to connect servers. The higher level switches are usually the bandwidth bottleneck and cost a lot. And they perform poorly in term of fault tolerance. Network equipment vendors have been investing great effort to flatten the data center networks as much as possible. Moreover, many new network architectures are proposed for the next-generation data centers network in academia, for example Fat-tree [1], VL2 [2], Dcell [3], BCube [4], CamCube [5]. Lucian et.al classify those designs into three broads, server-only, switch-only and hybrid architectures, based on the hardware equipment used to forward or relay network traffic [6]. In our work, we utilize a server-only 2D mesh interconnection network structure for data center interconnection. In the configuration, each server provides multiple network interfaces and plays dual roles, which are running regular applications and relaying traffics between servers [6]. In this way, no expensive high-end switches are needed.

Optical data center networks have great potential in improving power efficiency and growing bandwidth than traditional electronic based ones [7, 8]. The switching fabric in all-optical switching is usually the MEMS ones. And the switching time of optical switches or routers is always in the order of milliseconds. When they are utilized in server-centric data center networks, it is too long to employ them. No convenient optical buffers available also limit their utilizations. Electrical-optical hybrid switching is another choice for data center networks. Packets are treated in the form of electrical ones in the network interfaces. As soon as they reach output ports, they are converted to optical ones and transmitted on the fiber links. In our work, we employ a server model with combined input and output queuing and splitting corresponding O/E, E/O conversions.

## 2 Network scaling embarrassments

Our proposed mesh data center network employs a connection-oriented switching mechanism. Given arbitrary pair of servers in the network, S and D, there always exists a route from S to D. The complete route is called a path. Only one path is assigned to each pair of servers in a mesh data center network when dimension-order routing algorithm is employed.


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

Two types of paths pass through each server: a) current server is the initiator of a path; b) current server is en route of a path. Each server maintains a routing table that records the output port index of each path passing by the current server. When a packet arrives at a server, the routing message is stripped and sent to the internal arbitration unit. The arbitration unit simply looks up the routing table to discover the proper output ports and then switch the packet to it at appropriate occasions referring to the interior priority-based scheduling mechanism.

In our protocol, a maximum of $2k$ paths are conserved at each server in consideration of tradeoff between memory space and scalability. But as the data center network scales up, path number that each server needs record expands rapidly, and more memory is utilized to store additional paths. It can exceed the upper bound of path number easily. Therefore, it is unwise to record all the paths passing by a server. Hierarchical mesh network and different level subnet agents are employed to solve the scaling problem and obtain better performance.

## 3 Virtual hierarchical mesh network

No extra switches are utilized in the server-only mesh data center network. So, our proposed hierarchy mechanism is a virtual one on physical mesh network, which is different from the traditional hierarchical mesh network employing switches to converge traffics. However, the server addressing and routing algorithm are designed hierarchically. A three-level hierarchical mesh data center is shown to illustrate our proposed hierarchical mechanism in Fig. 1. But actually, there is no upper bound on the hierarchic number limit in our proposed hierarchical mesh network.

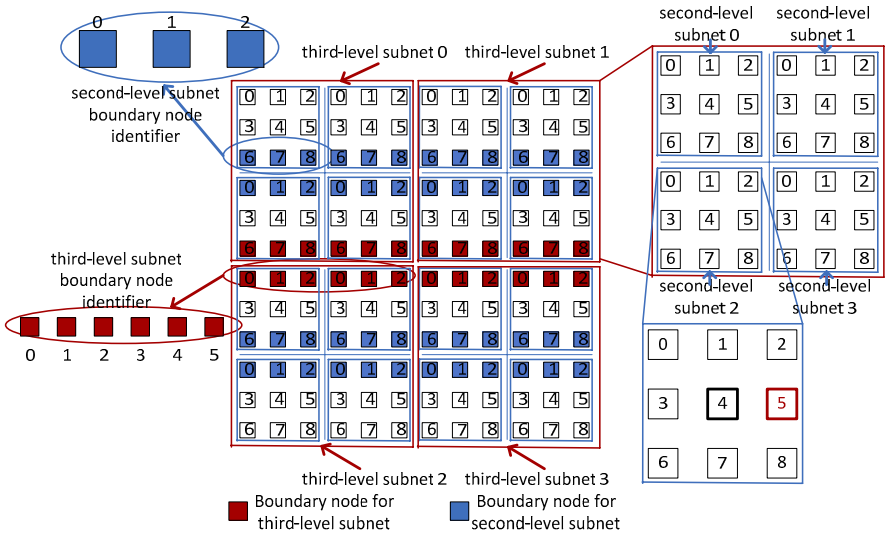


**Fig. 1.** A $12 \times 12$ virtual hierarchical mesh network

In Fig. 1, $12 \times 12$ servers compose of a small mesh data center network. Among them, a $3 \times 3$ sub-mesh is considered as a BC (basic cell) to construct higher level subnet network. $2 \times 2$ BCs form a third-level subnet and $2 \times 2$ third-level subnets cover the whole network. The size of BC is not restricted


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

as long as it is within the path number limit. Furthermore, a high level network is not necessarily a strictly symmetry mesh network. Arbitrary combinations of rows and columns are feasible.

### 3.1 Addressing of servers

Natural numbers are used for node address representation. For an n-level hierarchical mesh network, the address of each server is represented by an n-tuple $(a_n a_{n-1} \dots a_2 a_1)$, $a_i$ $(1 \leq i \leq n)$ is the address of level $i$ respectively. Each level address counts from 0 and adheres to row-major rule. As for the server which is highlighted in the BC of bottom right of Fig. 1, its address can be represented as (1, 2, 4).

### 3.2 Routing table construction and subnet agent selection

The $12 \times 12$ hierarchical mesh network in Fig. 1 is used to illustrate our routing table construction methods. The routing table of each server can be divided into three parts: the local routing table of BC that the current server belong to, the paths from the current server to another second-level subnet and paths from the current server to another third-level subnet. There are lots of servers which belong to another second or third level subnet. Servers, which are picked out to represent the corresponding level subnets, are named as subnet agents. Actually, paths from the current server to another second or third level subnet are paths from the current server to those agents.

Vast types and locations of agent can be obtained to guide the generation of each server's routing table. However, not all the selection methods are workable. Selections without considering agent location may lead to deadlock of the whole network. Servers at the far left of BC are selected to be agents of each level subnet. No turn exists from south or north to west, and it is deadlock-free naturally. But all possible 90° turns are allowed when the agents are selected at the middle or far right of BC. Cyclic dependence relationship is formed in the channel dependence graph of the mesh network and deadlock occurs. However, single-agent selection methods are not optimal for performance. Multi-agent selection method is another choice in Fig. 1. All the boundary servers in the same row of each level subnet are named as agents of the subnets they belong to. The servers filled with blue are agents of each second-level subnet, and the servers filled with red are that of each third-level subnet. Those boundary servers count from 0 and increase one by one. Each server still maintains those paths to agents of different level subnets. Additional paths are stored compared with our previous design. For a mesh network having $(k \times k) \times (m \times m) \times (n \times n)$ servers, maximum path number preserved by servers is formulated by the following expression:

$$Path_{total} = k^2mn + m^2n + n^3 - \frac{3 + (-1)^n}{2}n. \quad (1)$$

### 3.3 Hierarchical routing algorithm

A hierarchical routing algorithm is proposed to guarantee packets' efficient transmission. Routing of packets is performed from highest level to lowest


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

level. It is first done at the highest level network. Once a packet arrives at its highest level subnet, routing continues within a lower level subnet. The routing process is repeated until the packet arrives at its lowest level subnet finally. The destination address of the packet serves as routing message and is inserted into the header of the packet. When a packet arrives at a server, the routing unit simply strips the routing message and compares its destination address of each level subnet with that of the current server. Then, a routing table looking up operation is executed to obtain the right output port for transmission. Moreover, additional fields are defined in packets' headers. They are used to determine the exact positions of destination servers. As for the mesh network, third-level subnet index and second-level subnet index are added in packets' headers. Packets will be routed to servers which are in the same column as packets' destinations firstly. No additional hops are wasted before they arrive at their destinations.

Due to simplicity and fast routing, dimension-order routing algorithm is considered to configure each server's routing table once the whole network is powered up. When packets route in the multi-agents hierarchical mesh network, they always follow the dimension-order routing paths. Therefore the routing algorithm is minimal and no deadlocks occur. At the same time, path number each server conserves is considerably reduced. Compared with the single-agent method, link load presents much more balanced with all the links utilized.

## 4 Performance evaluations

All the proposed items are demonstrated and analyzed under the synthetic traffic patterns on an OPNET-based packet-level simulation platform developed by us. Each server in our models contains a five-port crossbar switch with combined input and output queuing. Customized credit-based flow control mechanism is employed for rate limiting and packets' lossless transmission. All links transmit packets at the rate of 10 Gbps. Packets have the uniform length of 256 bytes.

Fig. 2 shows a comparison of average end-to-end latency versus offered

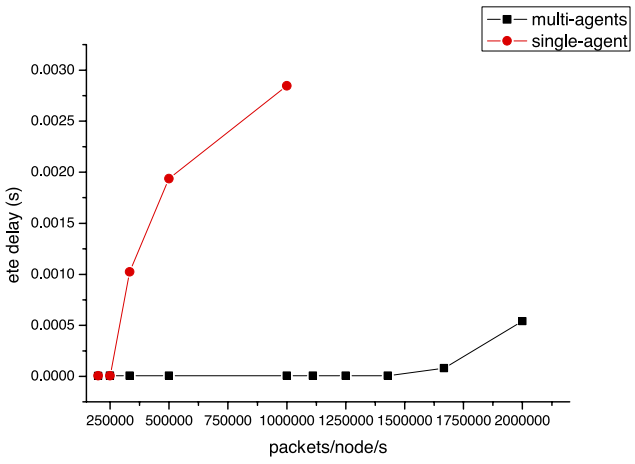


**Fig. 2.** Average end-to-end delay comparisons


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

traffic between single-agent and multi-agents selection methods under all-to-all traffic pattern. The average end-to-end delay of the single-agent increases sharply as the offered traffic increases. But the multi-agents method performs much better even in the case of heavy traffic. The average link utilizations of different location of the data center network with the average interarrival time of 0.000001 s are illustrated in Fig. 3. Fig. 3 (a) and (c) show the link utilization of horizontal and vertical links in the single-agent method while the Fig. 3 (b) and (d) show that of multi-agents method respectively. Compared with the single-agent method, more links are utilized to balance the network load with multi-agents method employed.

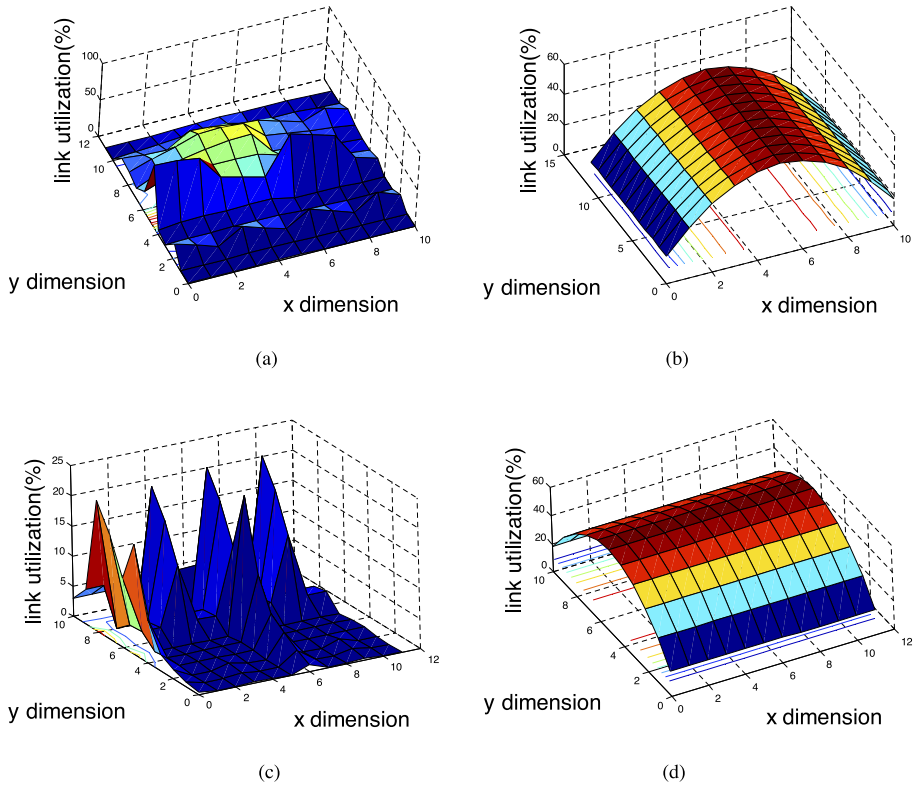


**Fig. 3.** Link utilization of different locations of the mesh network

## 5 Conclusion

This paper proposes a virtual hierarchical mesh network and hierarchical routing algorithm to reduce routing table size for better performance and scalability. Different agent selection methods are illustrated and tested to avoid network deadlock. The theoretical analysis and simulation results show that the multi-agents selection method outperforms the single-agent selection method in terms of average end-to-end latency, link throughput and load balance. As a future work, more balanced link load, fault-tolerance mechanisms and other topologies are being considered.


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012

## Acknowledgments

This work is supported partly by the National Science Foundation of China under Grant No.60803038, No.61070046, the special fund from State Key Lab (No.ISN1104001), the Fundamental Research Funds for the Central Universities under Grant No.K50510010010, the 111 Project under Grant No.B08038, and the Intel-University cooperation project.


DOI: 10.1587/elex.9.172
Received November 26, 2011
Accepted December 30, 2011
Published February 10, 2012