Load Balancing Mechanisms in Data Center Networks

Santosh Mahapatra Xin Yuan
Department of Computer Science, Florida State University, Tallahassee, FL 32306
{mahapatr,xyuan}@cs.fsu.edu

***Abstract*—We consider network traffic load balancing mechanisms in data center networks with scale-out topologies such as folded Clos or fat-tree networks. We investigate the flow-level Valiant Load Balancing (VLB) technique that uses randomization to deal with highly volatile data center network traffics. Through extensive simulation experiments, we show that flow-level VLB can be significantly worse than the ideal packet-level VLB for non-uniform traffics in data center networks. We propose two alternate load balancing mechanisms that utilize the real-time network state information to achieve load balancing. Our experimental results indicate that these techniques can improve the performance over flow-level VLB in many situations.**

I. Introduction

The emerging cloud computing paradigm has driven the creation of data centers that consist of hundreds of thousands of servers and that are capable of supporting a large number of distinct services. Since the cost of deploying and maintaining a data center is extremely high [5], achieving high utilization is essential. Hence, supporting *agility*, that is, the ability to assign any service to any server, is critical for a successful deployment of a data center.

Many networking issues in data center networks must be re-examined in order to support agility. In particular, the network must have layer-2 semantics to allow any server to be assigned to any service. In addition, the network must provide uniform high capacity between any pair of servers and support high bandwidth server-to-server traffics. Several proposals that can achieve these design goals have been reported recently [1], [4], [7]. All of these proposals advocate the use of scale-out topologies, such as folded Clos or fat-tree networks [3], to provide high bandwidth between servers.

Although fat-tree topologies have high bisection bandwidth and can theoretically support high bandwidth between any pair of servers, load balancing is essential for a fat-tree topology to achieve high performance. Balancing network traffic in a fat-tree based data center network poses significant challenges since data center traffic matrices are highly divergent and unpredictable [2]. The current solution addresses this problem through randomization [5], [7] using Valiant Load Balancing (VLB) [3] that performs destination independent random traffic spreading across intermediate switches.

VLB can achieve near optimal performance when (1) the traffics are spread uniformly at the packet level; and (2) the offered traffic patterns do not violate edge constraints. Under these conditions, the packet-level VLB technique is ideal for balancing network traffics and has been shown to have many nice load balancing properties [3]. However, although the fat-tree topology is well suited for VLB [4], there are restrictions in applying VLB in data center networks. In particular, to avoid the out-of-order packet issue in a data center network with TCP/IP communications, VLB can only be applied at the flow level (all packets in one flow use the same path) [4].

Given that data center traffics typically contain many large flows [2], it is unclear whether flow-level VLB can achieve similar performance as packet-level VLB, and whether flow-level VLB is more effective in dealing with traffic volatility than other load balancing techniques that utilize the network state information. These are the research questions that we try to answer. In this work, through extensive simulation experiments, we show that flow-level VLB can be significantly worse than packet-level VLB for non-uniform traffics in data center networks. We propose two alternate load balancing mechanisms that utilize the real-time network state information to achieve load balancing. Our experimental results indicate that these techniques can improve network performance over flow-level VLB in many situations.

The rest of the paper is structured as follows. Section II briefly introduces fat-tree based data center networks and the existing flow-level VLB load balancing mechanism. Section III compares the performance of flow-level VLB with that of packet-level VLB for different system configurations and traffic patterns. The results indicate that flow-level VLB can be significantly worse than packet-level VLB for non-uniform traffics. Section IV presents the proposed load balancing mechanisms that are based on real-time network state information. Section V studies the performance of the proposed schemes. Finally, Section VI concludes the paper.

II. Fat-tree based data center networks

We model fat-tree based data center networks based on VL2 [4]. A data center network consists of racks of servers. The servers in each rack are connected to a Top of Rack (ToR) switch. Each ToR switch is connected to several aggregation switches, which further connect to top tier intermediate switches. Fig 1 shows an example fat-tree based data center network. The system has four layers: the top layer intermediate switches, the second layer aggregation switches, the third layer top of rack switches, and the fourth level servers. Switch-to-switch links are typically faster than server-to-switch links.

In the example, server-to-switch links are 1Gbps and switch-to-switch links are 10Gbps. As shown in the figure, each aggregation switch has a bidirectional link connecting to each of the intermediate switches: such a 2-level folded Clos or fat-tree topology provides a richly-connected backbone and offers large aggregate capacity among servers. Each ToR switch is connected to two aggregation switches for load balancing and fault-tolerance purposes.

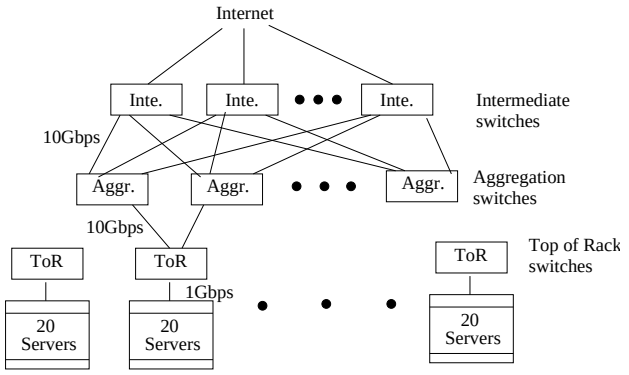

Fig. 1. An example fat-tree network between aggr. and intermediate switches

The topology of a fat-tree based data center network can be defined by the following parameters: the number of ports in intermediate switches (d_i), the number of ports in aggregation switches (d_a), the number of servers per rack (npr), the number of aggregation switches that each ToR switch connects to (u_{tor}), the switch-to-switch bandwidth (bw_{sw}), and the server-to-switch bandwidth (bw_{ms}). Given these parameters, one can derive the following: the ratio of the switch-to-switch bandwidth to server-to-switch bandwidth $R = \frac{bw_{sw}}{bw_{ms}}$; the number of intermediate switches is $\frac{d_a}{2}$; the number of aggregation switches is d_i; the number of ToR switches is $\frac{d_i \times \frac{d_a}{2}}{u_{tor}}$; the number of servers is $\frac{d_i \times \frac{d_a}{2}}{u_{tor}} \times npr$; the numbers of links between intermediate switches and aggregation switches, and between aggregation switches and ToR switches are $\frac{d_a}{2} \times d_i$; the number of links between ToR switches and servers is $\frac{d_i \times \frac{d_a}{2}}{u_{tor}} \times npr$. Hence, in a *balanced* design where the links in all levels in the system provide the same aggregate bandwidth, we have

$$\frac{d_i \times \frac{d_a}{2}}{u_{tor}} \times npr = \frac{d_a}{2} \times d_i \times R \Rightarrow npr = R \times u_{tor}.$$

Hence, we will call a system with $npr = R \times u_{tor}$ a *balanced* system. When $npr > R \times u_{tor}$ ($\frac{d_i \times \frac{d_a}{2}}{u_{tor}} \times npr > \frac{d_a}{2} \times d_i \times R$), we will call such a system an *over-subscribed* system; when $npr < R \times u_{tor}$ ($\frac{d_i \times \frac{d_a}{2}}{u_{tor}} \times npr < \frac{d_a}{2} \times d_i \times R$), we will call such a system an *under-subscribed* system. When the system is fully loaded, in an over-subscribed system, upper-level links cannot handle all traffics generated by the servers. While new designs use balanced systems to support high capacity server-to-server traffics [1], [4], over-subscription is common in practical data center networks for cost-saving purposes [5].

A. Valiant Load Balancing (VLB)

In the traditional packet-level VLB, routing consists of two stages. In the first stage, the source node splits its traffic randomly to an intermediate node. The intermediate node is randomly selected for each packet and does not depend on the destination. In the second stage, the intermediate node forwards the packet to the destination. Valiant load balancing, which is commonly known as two-stage load balancing, is an oblivious routing scheme that achieves close to optimal performance for arbitrary traffic matrices [6], [8].

The fat-tree topology is well suited for VLB in that VLB can be achieved by indirectly forwarding traffic through a random intermediate switch: packet-level VLB can provide bandwidth guarantees for any traffic matrices that satisfy the edge constraint [8]. As discussed earlier, the main issue is that in a TCP/IP network, VLB can only be applied at the flow level to avoid the out-of-order packet issue that can cause performance issues in TCP. Flow-level VLB can have performance problems: when large flows are present, the random placement of flows can potentially lead to persistent congestion on some links while others are under-utilized. Since a fairly large percentage of flows in data center traffics are large [2], this may be a significant problem.

III. Performance of Flow-level VLB

In this section, we report our study of the relative performance of flow-level VLB and packet-level VLB. To study the performance, we develop a data center network simulator that is capable of simulating data center networks of different configurations. The network topologies are specified using the following parameters: the number of ports in intermediate switches (d_i), the number of ports in aggregation switches (d_a), the number of servers per rack (npr), the number of aggregation switches that each ToR switch connects to (u_{tor}), the switch-to-switch bandwidth (bw_{sw}), and the server-to-switch bandwidth (bw_{ms}). The bandwidth ratio $R = \frac{bw_{sw}}{bw_{ms}}$. The simulator supports many traffic patterns including random uniform traffics and random non-uniform traffics, it simulates TCP (including connection management and TCP flow control and congestion control mechanisms) as the underlying communication mechanism with the assumption of no transmission errors and infinite buffers in switches. In studying packet-level VLB, we assume that the end-points can buffer all out-of-order packets and do not cause TCP packet retransmissions when out-of-order packets arrive.

We use average *packet latency* as the performance metric to evaluate the schemes. A better load balancing scheme will result in a smaller average packet latency. Each of the (random) experiments is repeated 32 times. We compute the mean packet latency and 95% confidence interval based on the 32 random samples. For all experiments that we perform, the 95% confidence intervals are less than 2% of the corresponding mean values, which indicates that the mean values obtained are with high confidence levels. Thus, we will report the mean values and use the mean values to derive other metrics such as the performance improvement percentage.

In the experiments, we find that flow-level VLB offers similar performance to packet-level VLB in many situations, for example, when the traffic patterns are uniform random traffics, when the system is under-subscribed, and when the message size is not very large. Fig. 2 and Fig. 3 show two representative cases for uniform traffic patterns. In these patterns, each source-destination pair has a uniform probability to communicate (or not to communicate). The message size for each communication in the experiments is 1Mbits. Fig. 2 shows the case for a balanced system with $d_i = 8$, $d_a = 8$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$, while Fig. 3 shows the case for an over-subscribed system with $d_i = 8$, $d_a = 8$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 10Gbps$, $bw_{ms} = 10Gbps$. As can be seen from the figures, the performance of flow-level VLB and packet-level VLB is very similar for both cases.

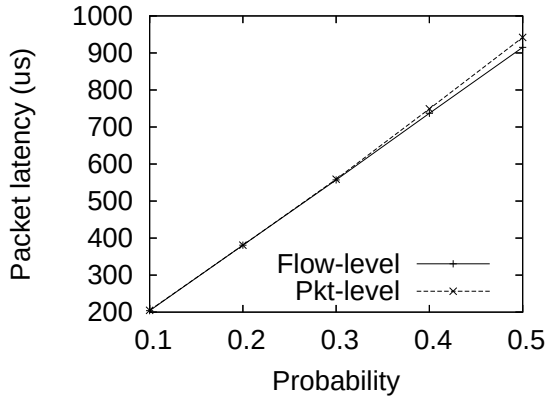

Fig. 2. Performance for uniform traffic patterns on a balanced system ($d_i = 8$, $d_a = 8$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

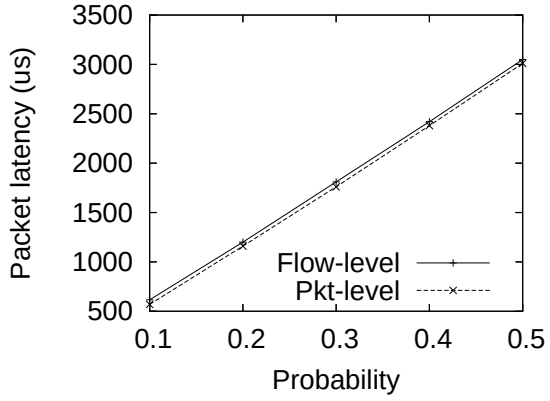

Fig. 3. Performance for uniform traffic patterns on an over-subscribed system ($d_i = 8$, $d_a = 8$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 10Gbps$, $bw_{ms} = 10Gbps$)

While flow-level VLB performs well on uniform traffics, it does not deliver high performance for non-uniform traffics. Fig. 4 shows the improvement percentage of packet-level VLB over flow-level VLB for random clustered traffics on different sized balanced systems. The clustered traffic is generated as follows. First, the servers in the system are randomly partitioned into clusters of a certain size. After that, communications are only performed between servers in the same cluster. This random clustered traffic pattern will be used throughout this paper as a representative non-uniform traffic pattern. In the experiments, the cluster size is 3 and the message size is 400000 bits. Other parameters are $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$. The numbers of ports in intermediate switches and aggregation switches are the same and vary as shown in the figure to allow different sized systems to be investigated. For example, the system with $d_i = d_a = 24$ supports $24 \times 12 \times 8 = 2304$ servers, which are partitioned into 768 3-server clusters. As can be seen in the figure, for the random clustered patterns, packet-level VLB is noticeably better than flow-level VLB, and the performance gap is larger as the system size increases.

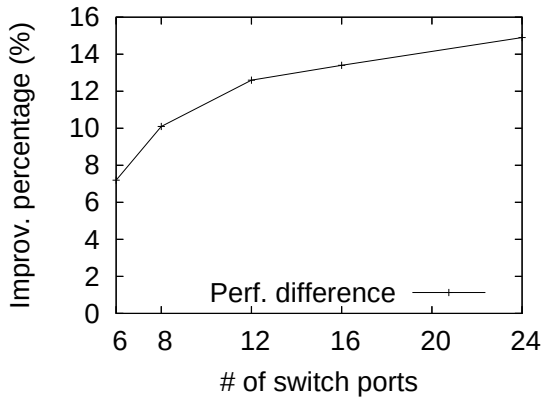

Fig. 4. Improvement percentage of packet-level VLB over flow-level VLB for clustered traffics on different sized balanced systems ($u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

The results in Fig. 4 are for the relatively small message size of 400000 bits. Fig. 5 shows the impact of message sizes. Other parameters used in this experiment are: $d_i = d_a = 12$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$, and $cluster\ size = 4$. As the message size increases, packet-level VLB becomes more effective in comparison to flow-level VLB. Flow-level VLB has performance issues when it routes multiple flows into one link and causes congestion. The congestion will not be resolved until the messages are completed. When the message size is larger, the system becomes more congested for a longer period of time and the packet delay increases. Such a problem will not occur with packet-level VLB where each packet is routed randomly. This result indicates that flow-level VLB can have serious problems in practical data center networks with many large flows and each large flow easily having more than 100MB data [4]. Notice that using average packet delay as the performance metric can somewhat under-estimate the severity of congestion since packets that are not congested are also counted in the average.

Fig. 6 shows the impact of the number of servers per rack. Other parameters in the experiments are: $d_i = d_a = 12$, $u_{tor} =$

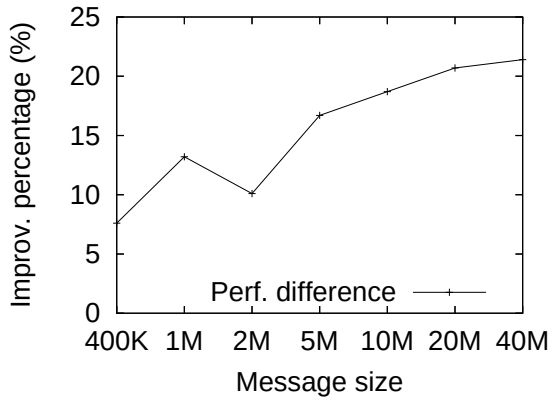

Fig. 5. Impact of message size ($d_i = d_a = 12$, $u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

2, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$, and $cluster\ size = 3$. With the same $R = 40/10 = 4$ in the experiments, when $npr = 8$, the system is balanced; when $npr < 8$, the system is under-subscribed; when $npr > 8$, the system is over-subscribed. When the system is under-subscribed, flow-level VLB has the same performance as packet level VLB. This is because higher level links are not congested with either packet-level or flow-level VLB scheme. When the system is balanced or over-subscribed, effective load balancing becomes more critical, and packet-level VLB performs better than flow-level VLB. It is interesting to see that flow-level VLB has the worst relative performance when the system is balanced.

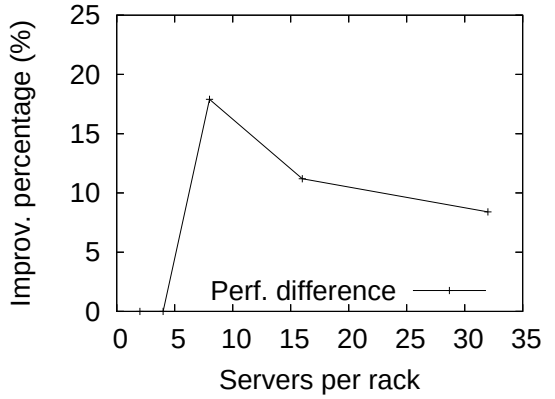

Fig. 6. Impact of the number of servers per rack ($d_i = d_a = 12$, $u_{tor} = 2$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

Fig. 7 shows the performance for different bandwidth ratios on balanced systems. In this experiment, npr increases with R to maintain system balance. Other parameters are: $d_i = d_a = 16$, $u_{tor} = 2$, $bw_{ms} = 1Gbps$, message size is 1000000 bits, cluster size is 3. Flow-level VLB is significantly worse than packet-level VLB when R is small. The performance gap decreases as R increases. This is because for a small R, a small number of servers can saturate the upper level links and the chance for this to happen is higher than the chance for many servers to saturate an upper level link in cases when R is large. These results indicate that flow-level VLB can be effective for data center networks with large bandwidth ratios.

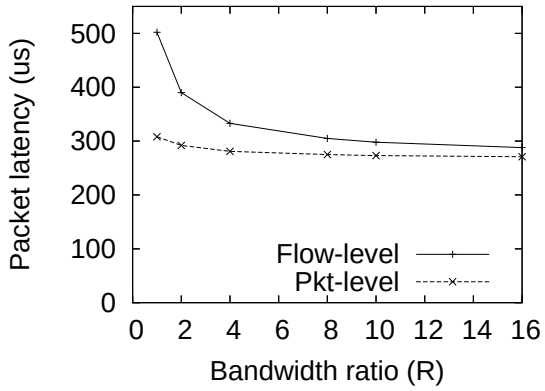

Fig. 7. Performance for different bandwidth ratios (R) ($d_i = d_a = 16$, $u_{tor} = 2$, $bw_{sw} = 1Gbps$, $msize = 1000000$)

In summary, flow-level VLB works well for many situations such as uniform random traffics, under-subscribed systems and relatively small message sizes. It does not provide high performance for non-uniform traffics, especially for large messages, large systems, or small bandwidth ratios. These results motivate us to search for alternative load balancing mechanisms that can achieve high performance in these situations.

IV. Proposed Load Balancing schemes

We propose two alternate load balancing schemes to overcome the problems in the flow-level VLB. Instead of relying on randomization to deal with traffic volatility, our schemes explicitly take the link load information into consideration. Since the hardware and software for data center networks are still evolving, we do not limit our design by the current technology constraints; and some of our assumptions may not be supported by the current hardware. Our first scheme is called *queue-length directed adaptive routing*. This scheme achieves load balancing by routing flows based on the queue-lengths of the output ports of switches at the time when the flow starts communicating. The second scheme is called *probing based adaptive routing*, which probes the network before transmitting a large flow.

A. Queue-length directed adaptive routing

The key objective of any load balancing scheme is to direct traffic to avoid congested links. The queue-length directed adaptive routing scheme adopts the adaptive routing scheme in high performance interconnects to data center networks. It works by selecting the output port with the smallest queue length that can reach the destination. The first packet of a flow (e.g. the SYN packet of a TCP flow) adaptively constructs the path towards the destination based on the queue-lengths of output ports. Once the path is determined, all other packets will follow the same path to avoid the out-of-order packet problem. Like flow-level VLB, the load balancing granularity

of our scheme is also at the flow level. This scheme guarantees that at the time when a flow starts, the least load links will be selected to form the path. This scheme is similar to traditional adaptive routing schemes that are used in high performance interconnects, which are proven technology. The main difference between our scheme and the traditional scheme is that the granularity of adaptivity is at the flow level in our scheme, which makes it more suitable for TCP/IP communications.

B. Probing based adaptive routing

This scheme assumes that the source node (either the server or the network interface card) can decide the path of a packet and send a probing packet following the path. This can be done with the traditionally source routing scheme. In probing based adaptive routing, when the source node needs to send a large flow, it first probes the network by sending a number of probe packets following different paths to the destination. The number of paths to be probed is a parameter of the scheme. The receiving node replies with an acknowledgment packet for the first probe packet received and drops the other probe packets for the flow. The acknowledgment packet carries the path information, and the source will then use the selected path for the communication.

Probing the network information for large flows ensures that large flows are not routed over congested links, and thus achieves load balancing. This scheme allows the current network state to be probed before a large flow is routed and decreases the chance of network congestion. Note that probing based adaptive routing can be built over the traditional TCP/IP communication mechanism without additional support as long as the system supports source routing.

V. Performance of the proposed schemes

This section reports our performance study for the proposed load balancing schemes. The experimental setting in this section is similar to that in Section III. We first compare the performance of the two proposed schemes. These two schemes have very similar performance in all the experiments that we performed with different traffic patterns and system configurations. Fig. 8 and Fig. 9 show two representative cases. Fig. 8 has the same system configurations and traffic patterns as those for Fig. 4; and Fig. 9 has the same system configurations and traffic patterns as those for Fig. 7. In the probing based scheme, 4 paths are probed to select the path for each flow. As in many other experiments that we carried out, the two proposed schemes almost have the same packet delay. The results are not unexpected since both schemes use the network state information to direct traffic to underloaded links with different mechanisms. Fundamentally, both queue length directed adaptive routing and probing based adaptive routing use the same heuristic, that is, scheduling each flow to the least congested links, and have a very similar resulting performance. We note that queue-length directed adaptive routing will require more system support than probing based adaptive routing. Since these two schemes have similar performance, we will only report the results for the probing based adaptive routing scheme. In the experiments in the rest of this section, probing based adaptive routing probes 4 paths to select the path for each flow.

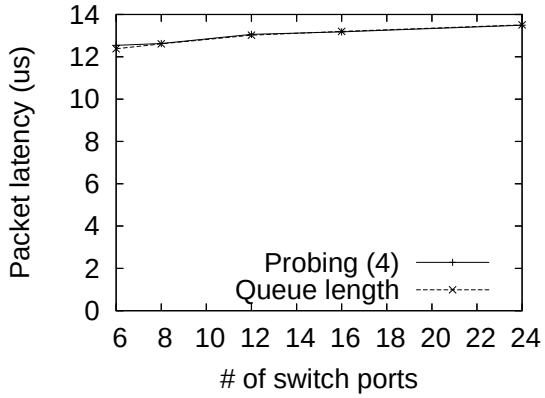

Fig. 8. Performance of the proposed load balancing schemes for clustered traffics on different sized balanced systems ($u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

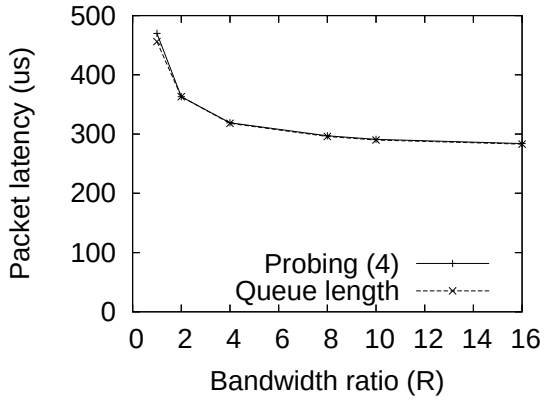

Fig. 9. Performance of the proposed load balancing schemes for clustered traffics on balanced systems with different bandwidth ratios ($d_i = d_a = 16$, $u_{tor} = 2$, $bw_{ms} = 1Gbps$, msize=1000000)

Fig. 10 shows the performance improvement percentage of the probing based adaptive routing scheme over flow-level VLB for clustered traffics on balanced systems. The traffic patterns and system configurations are the same as those for Fig. 4. For different network configurations, the probing based adaptive routing scheme consistently improves the packet latency by about 4%.

Fig. 11 shows the performance improvement of probing based adaptive routing over flow-level VLB for clustered traffics on balanced systems with different bandwidth ratios. The traffic patterns and system configurations are the same as those for Fig. 7. As can be seen from the figure, the probing based scheme consistently offers better performance over flow-level VLB for different configurations with better improvements for smaller R's. Fig. 12 shows the relative performance of flow-level VLB, the probing based scheme, and packet-level VLB for the same configuration. Packet-

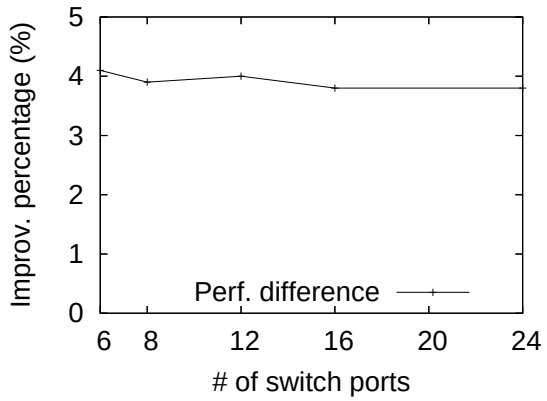

Fig. 10. Improvement percentage of probing based adaptive routing over flow-level VLB for clustered traffics on different sized balanced systems ($u_{tor} = 2$, $npr = 8$, $bw_{sw} = 40Gbps$, $bw_{ms} = 10Gbps$)

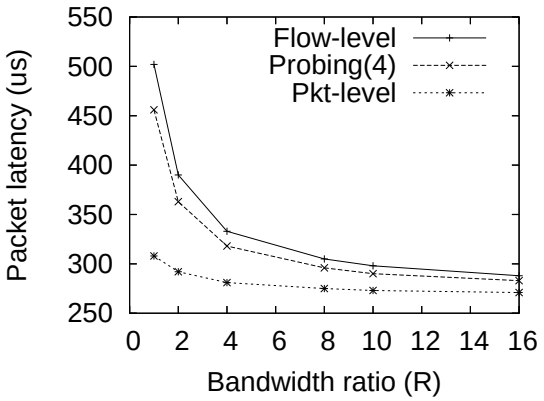

Fig. 12. Packet delays for clustered traffics on balanced systems with different bandwidth ratios ($d_i = d_a = 16$, $u_{tor} = 2$, $bw_{ms} = 1Gbps$, msize=1000000)

level VLB significantly out-performs the other two flow-level schemes in all configurations.

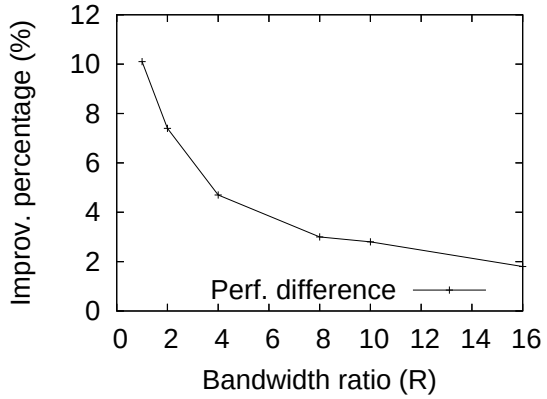

Fig. 11. Improvement percentage of probing based adaptive routing over flow-level VLB for clustered traffics on balanced systems with different bandwidth ratios ($d_i = d_a = 16$, $u_{tor} = 2$, $bw_{ms} = 1Gbps$, msize=1000000)

The proposed schemes achieve better performance than flow-level VLB for most situations where flow-level VLB performs noticeably worse than packet-level VLB. This indicates that the proposed schemes can be good complements to flow-level VLB since they can better handle the situations when flow-level VLB is not effective. However, packet-level VLB performs significantly better than the proposed schemes. This is due to the inherent limitation of flow-level load balancing schemes that require all packets in a flow to follow the same path, which limits the load balancing capability of the techniques.

VI. Conclusion

We investigate techniques for balancing network traffic in fat-tree based data center networks. We demonstrate that flow-level VLB can be significantly worse than packet-level VLB in balancing loads. We propose two methods, queue-length directed adaptive routing and probe-based adaptive routing, that explicitly consider the real-time network condition. The simulation results indicate that both schemes can achieve higher performance than flow-level VLB in the cases when flow-level VLB is not effective. The results also indicate that flow-level load balancing schemes include our newly proposed schemes can be significantly worse than packet-level VLB, which raises a question: how effective any flow-level load balancing scheme can be in data center networks with many large flows?

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable Commodity Data Center Network Architecture," *ACM SIGCOMM'08*, pages 63-74, August 2008.

[2] T. Benson, A. Anand, A. Akella, M. Zhang, "Understanding Data Center Traffic Characteristics," *ACM SIGCOMM Computer Communication Review*, 40(1):92-99, Jan. 2010.

[3] W. J. Dally and B. Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann Publisher, 2004.

[4] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, S. Sengupta, "VL2: A Scalable and Flexible Data Center Networks," *ACM SIGCOMM'09*, pages 51-62, August 2009.

[5] A. Greenberg, J. Hamilton, D. A. Maltz, P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Communication Review*, 39(1):68-73, Jan. 2009.

[6] M. Kodialam, T. V. Lakshman,and S. Sengupta, "Maximum Throughput Routing of Traffic in the Hose Model," *IEEE Infocom*, 2006.

[7] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, A. Vahdat, "PortLand: a Scalable Fault-Tolerant Layer 2 Data Center Network Fabric," *ACM SIGCOMM'09*, pages 39-50, August 2009.

[8] R. Zhang-Shen and N. McKeown, "Design a Predictable Internet Backbone Network," In *Thirteenth International Workshop on Quality of Service*, 2005.