ICIC Express Letters
Volume **10**, Number **12**, December **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2881–2889**

MEASURING THE DIFFERENCE OF PACKET LATENCIES IN DATA CENTER

Jiehua Wang[1], Xiaohui Zhu[1,2] and Surong Chen[1]

[1]School of Computer Science and Technology
Nantong University
No. 9, Seyuan Road, Nantong 226019, P. R. China
{ wang.jh; zhufirst; firstemail }@ntu.edu.cn; ntuzxh@hotmail.com

[2]Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University
No. 111, Ren'ai Road, Suzhou Industrial Park, Suzhou 215123, P. R. China

Received April 2016; accepted July 2016

Abstract. *More and more applications are being moved to data center. However, little research for the characteristic of the difference of packet latencies in data center was carried out. Packet latency is a key issue of latency-sensitive applications. In this paper, we explore the nature of packet latency in a 3-level tree topology data center and make a detailed study of the difference of packet latencies by using NS2 to analyze the packet-level traces with different data transmission protocols. We find that though the workload and over-subscription in data center have a significant effect on the difference of packet latencies for most of the data transmission protocols, pFabric has a better performance than other protocols and the difference of packet latencies is much smaller than others. We also use ELR (equal-length routing) to equalize the pathway length for all packets and find that the difference of pathway length has little impaequal-length routingct to the difference of packet latencies on regular TCP protocols, but has a significant effect on pFabric. We also verify that pFabric with ELR has a higher performance than other data transmission protocols for the difference of packet latencies and can meet the requirements of most real-time applications.*

Keywords: Packet latency, Data center, pFabric, Data transmission protocol, Equal-length routing

1. **Introduction.** A data center refers to a large, dedicated cluster of computers [1]. More and more companies and organizations are moving their applications to private or public data centers. However, for the reason of security, reliability or others, some applications such as stock trading system, and real-time bidding system are not suitable to be deployed in public data centers. So some companies are building their own private data centers to run their own latency-sensitive applications. Assume that several customers are bidding an auction product at the same time using a real-time bidding system in a data center. Customers A and B both submit their requests with the same bidding price back to back. Let R_A be the bid request of customer A and R_B be the request of customer B. R_A is a little earlier than R_B. Because R_A and R_B are submitted from different source hosts and arrive at the same destination server, it is obvious that the packet latencies from source host to the destination server for these two packets are different. If packet latency for R_A is much higher than for R_B, it will be customer B who can bid the auction product successfully. However, it is obviously unfair for customer A, because Customers A and B give the same price for the auction product and customer A submits the request earlier than customer B. So customer A should win this auction. With the development of data center, some researchers are focusing on analyzing the data transmission characteristic

in data center [2,3] and several new data transmission protocols have been proposed in recent years.

DTCP: DTCP [4] tries to use ECN [5] and less buffer space to improve the workload throughput, burst tolerance and low latency for short flows.

HULL: HULL [6] can significantly reduce average and tail latency by sacrificing a small amount of bandwidth.

PDQ: PDQ [7] enables flow preemption to approximate a range of scheduling disciplines by using SYN and FIN to perform the required book-keeping.

DRB: DRB [8] uses a per-packet round-robin based routing algorithm to achieve perfect packet interleaving resulting in both high bandwidth utilization and low latency.

pFabric: pFabric [9] can provide near theoretically optimal FCT (flow complete time) for both short and long flows. In pFabric, each packet is set a priority value indicating the packet's priority in switch queues. The queue uses priority-based scheduling/dropping algorithm to determine which packet should be re-transmitted or dropped.

However, we still have no idea whether these protocols can meet the requirement for latency-sensitive applications. To the best of our knowledge, it is the first effort to give a deep look at the difference of packet latencies in data center and try to find a way to minimize the difference of packet latencies.

Motivated by this observation, we carry out this research on packet latency in data center to find if the difference of packet latencies in data center has great effect on these applications and how we can minimize the difference of packet latencies to promote the fairness for all the customers. We study the packet latency by simulating the actual data transmission with a 3-level tree topology and different data transmission protocols in NS2 [10]. We try to answer these questions:

- What is the packet latency in a 3-level tree topology data center with different background workloads and over-subscriptions?
- What kind of factors has a significant effect on the difference of packet latencies in the data center?
- Which data transmission protocol has the best performance in minimizing the difference of packet latencies in the data center?
- Can we further minimize the difference of packet latencies?

The rest of this paper is structured as follows. Section 2 describes the background information of FullTcp and pFabric. In Section 3, we describe the detailed simulation results and analysis. At last, we summarize our findings and conclusions in Section 4.

2. Background.

pFabric implements a priority-based dropping/scheduling mechanism for switch queues. The extensive simulations in NS2 showed that pFabric can provide a near-optimal performance for both short and long flows and it is one of the best data transmission protocols in recent-proposed brand-new protocols. Here, we try to verify if pFabric also has a good performance in the difference of packet latencies and can meet the special requirement for real-time applications. As regular TCP protocols are widely used in today's data center, we use FullTcp as a baseline to compare to pFabric. Considering many data centers are using 3-level tree topology [11,12], we mainly focus on a 3-level tree topology with 1 core switch, 2 aggregation switches, 8 top of rack switches (TOR switches) and 128 servers in this paper. Assume each TOR switch has the same amount of servers, so the over-subscription for servers, TOR switches and aggregation switches is $64 : 4 : 1$. We set the link delay time of 0.2μs, host server delay time of 2.5μs, packet size of 1,500 bytes and bandwidth between host servers to TOR switch of 1Gbps. So we can calculate that the packet latency in TOR switch queue is 12μs ($1500 * 8/1\text{Gbps} = 12\mu\text{s}$). Let $V = \{V_0, V_1, \ldots, V_{127}\}$ be the collection of 128 host servers. The detailed topology is shown in Figure 1.

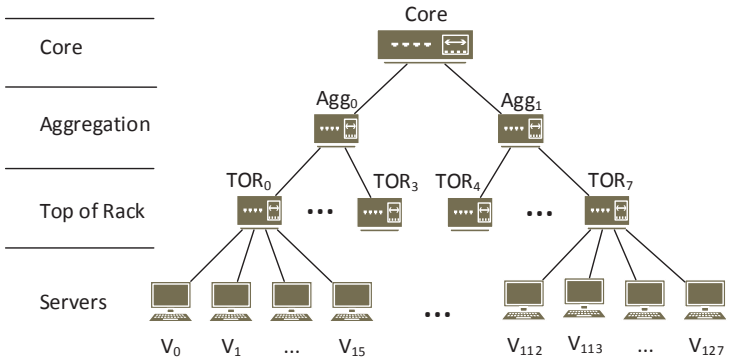


FIGURE 1. A 3-level tree topology

3. **Simulations and Evaluation.** For the sake of realism and the comparison between FullTcp and pFabric, we use the same empirical traffic distribution model [13], which is combined with small and large flows both for pFabric and FullTcp as background workload. In our simulations, every server sends a data flow to all the other servers in the topology, which means there are 16,256 different flows ($127 * 128$) at one time. These flows consist of the whole background workloads in our simulations.

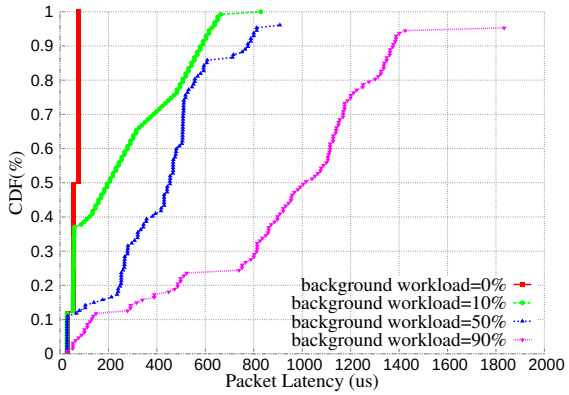
3.1. **Simulation for regular TCP.** We do simulations in NS2 as the following steps.

- First, we create 127 test flows from server V_0 to all the other servers without any background workload. Each test flow only sends one packet back to back.
- Second, we create the same 127 flows as the first step along with a 10% background workload.
- Third, we create the same 127 flows with a 50% dynamical background workload.
- Fourth, we create the same 127 flows with a 90% dynamical background workload.
- Fifth, we trace and collect all the packet latencies for these 127 test flows with different workloads and over-subscription and get the mean and standard deviation of packet latencies.

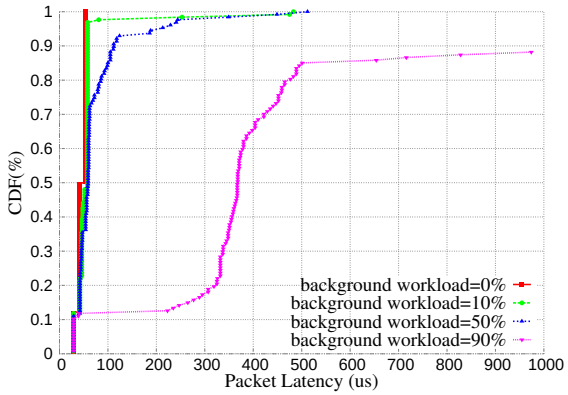
The detailed latencies for these 127 test packets are illustrated in Figure 2.

First, we set the bandwidth to 1Gbps both for core and aggregation switches resulting in a $64 : 4 : 1$ over-subscription topology. Figure 2(a) shows all the packet latencies for 127 test packets with different background workloads of 0%, 10%, 50% and 90%. We observe that the back-to-back latencies of these 127 test packets still vary from 29μs to 78μs even there is no background workload. This is because different packets are transmitted through different routes and switches with different lengths of route. We can learn from Figure 1 that the first packet set $(P_0, P_1, \ldots, P_{14})$ only crosses the TOR_0 switch and the latency is about 29μs. The second packet set $(P_{15}, P_{16}, \ldots, P_{63})$ crosses TOR_0, Agg_0 and TOR_1 (or TOR_2, TOR_3) switches and the latency is about 54μs. The latency for the third packet set $(P_{64}, P_{65}, \ldots, P_{126})$ is about 78μs. When background workload is increased to 10%, the packet latencies are also increased and vary from 29μs to 829μs. This is because influenced by background workloads, some test packets will spend more time waiting for transmission, which evidently increases their packet latencies. We get the similar results when we increase the background workload to 50% and 90%. Figure 2(a) shows that the background workload has a significant effect on packet latency.

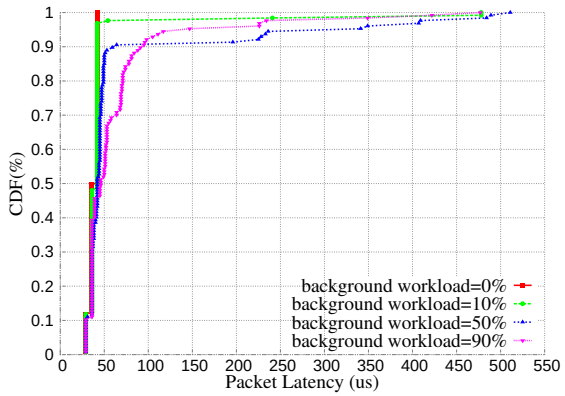
Second, we set the bandwidth to 2Gbps both for core and aggregation switches to consist of an over-subscription of $32 : 4 : 1$. Figure 2(b) shows that compared to Figure 2(a), packet latencies for test packets are obviously declined. This is because when we increase the bandwidth capacity both for core and aggregation switches, packets are transmitted



(a) Latency with an oversubscription of 64 : 4 : 1



(b) Latency with an oversubscription of 32 : 4 : 1



(c) Latency with an oversubscription of 16 : 4 : 1

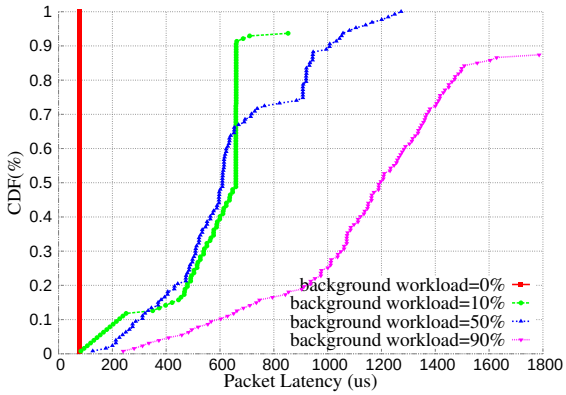
FIGURE 2. Packet latency for regular Tcp with different oversubscriptions

more quickly and less time is spent waiting for transmission in switch queues, which significantly reduces packet latency.

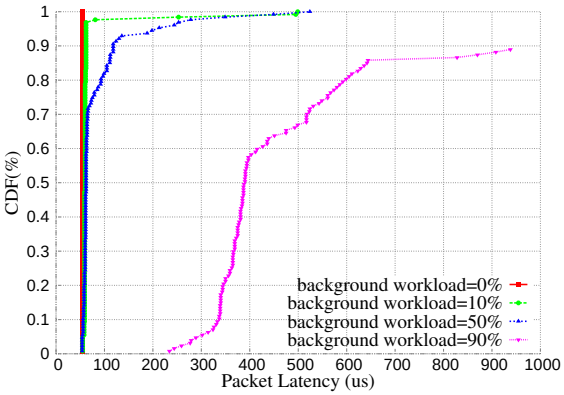
Third, when we increase the bandwidth capacity to 4Gbps both for core and aggregation switches resulting in an over-subscription of 16 : 4 : 1, packet latencies are significantly reduced. According to Figure 2, we argue that packet latencies can be significantly affected by background workloads. However, when we reduce the over-subscription, we can correspondingly decrease the negative impact of background workloads.

In order to have a deep look at how far the over-subscription and background workload affecting the difference of packet latencies, we get the mean and standard deviation of packet latencies on regular TCP in Figures 6 and 7. We find that when over-subscription is 64 : 4 : 1 and workload is 0%, the mean packet latency is 63.1μs (Figure 6(a)) and the standard deviation of packet latencies is 16.7μs (Figure 7(a)). When we increase the background workload, the mean and stand deviation of latencies are also increased. This is because higher workload spends more bandwidth and switches have to transmit more packets which will let some test packets spend more time for waiting in switch queues. In general, higher over-subscription and workload result in higher packet latency and higher difference of packet latencies.

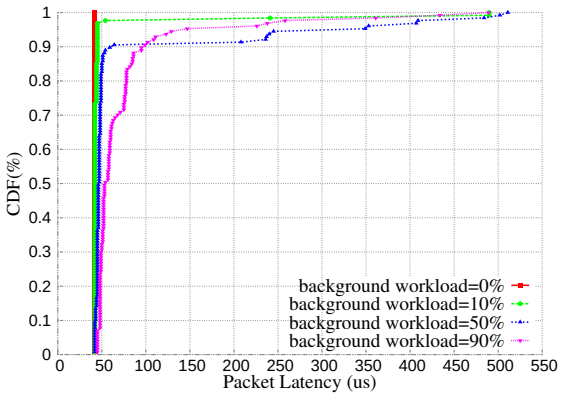
According to Figures 2, 6 and 7, we observe that on the one hand, these 127 test packets have different packet latencies even there is no background workload. On the other hand, the background workload and over-subscription have a significant effect on the difference of packet latencies. Based on these analyses, we make a conclusion that regular TCP protocol has a bad performance in minimizing the difference of packet latencies in data center.



(a) Latency with an oversubscription of 64 : 4 : 1



(b) Latency with an oversubscription of 32 : 4 : 1



(c) Latency with an oversubscription of 16 : 4 : 1

FIGURE 3. Packet latency for regular TCP with ELR

3.2. **Simulation for regular TCP with equal-length routing.** In order to understand how far the difference of pathways affects the difference of latencies, we use equal-length routing (ELR) to manually let all the test packets cross the core switch to have the same length of pathways. We do the same simulations as in §4.1 with the same configuration and topology except for the additional ELR and get the simulation results shown in Figure 3.

We set the over-subscription of 64 : 4 : 1 without any background workload (Figure 3(a)), the latencies for all test packets are as same as 78μs. This is because all the test packets have the same length of routing and cross the same amount of switches. We can find similar results from Figures 3(b) and 3(c). In Figure 3(a), when the background workload is increased to 10%, the packet latencies vary from 83μs to 853μs with 8 packets lost. This is because though these test packets have the same length of pathway, they cross different switches with different routes. At a certain time, each switch has a different status in its queue and some switches may have less packets in queue waiting for transmission than others, which leads to different latencies for different test packets. When we increase the background workload to 50% and 90% separately, we can see the similar results.

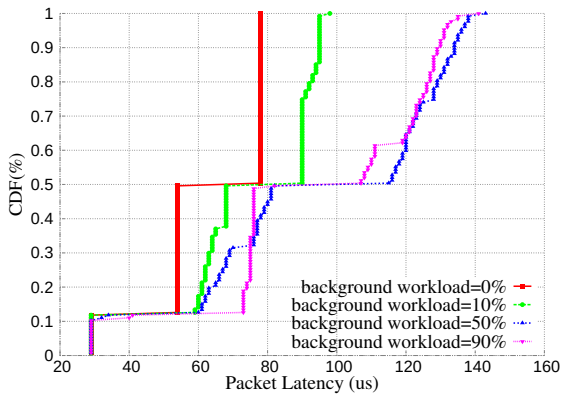
Comparing Figure 2 to Figure 3, we observe that ELR cannot significantly minimize the difference of packet latencies when data center has background workload. We also get the mean and standard deviation of packet latencies with different over-subscriptions and background workloads shown in Figures 6 and 7. According to the simulation results both for regular TCP and regular TCP with ELR in Figures 6 and 7, we can know that in regular TCP protocol, the length of packet pathway in data center has little effect on the difference of packet latencies. The difference of packet latencies is mainly determined by

the background workload and over-subscription. In general, higher background workload and over-subscription have higher difference of packet latencies in data center.

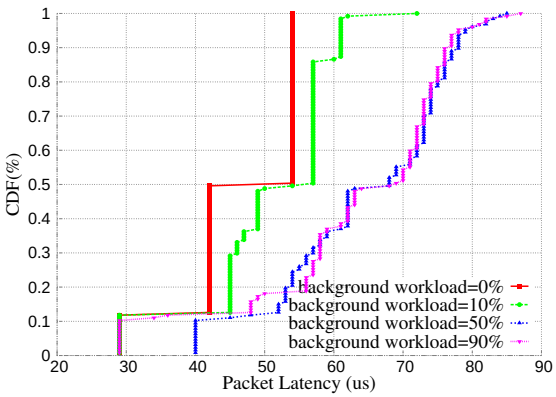
3.3. **Simulation for pFabric.** pFabric gave the short flows higher priority than long flows and used priority-based packet scheduling/dropping mechanism on each switch queue, which can significantly reduce the FCT (flow completing time) [9,12] both for short and long flows. we use the same 3-level tree topology and same parameters in pFabric as Figure 1 shows. Figure 4 shows the simulation results.

We can know from Figure 4(a), when there is no background workload, the packet latencies are just as same as in regular TCP and vary from 29μs to 78μs. When we increase the background workload to 10%, 50% and 90% separately, the latencies also increase correspondingly. Similar to §4.1, when we set the over-subscription of $32:4:1$ and $16:4:1$ separately, the packet latencies are also correspondingly decreased (Figures 4(b) and 4(c)).

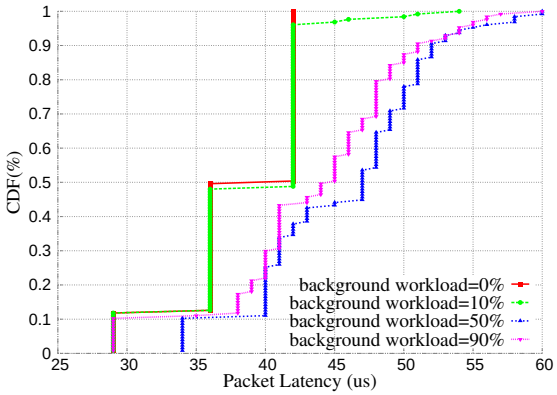
Comparing Figure 4 to Figure 2, we find the packet latencies in pFabric are much smaller than in regular TCP. This is because these test flows only have one packet to transmit, which means all the test packets have higher priority in pFabric than most of the background flow packets and are transmitted quickly in switch queues even with a high background workload. As a result, there is no packet lost in pFabric and their packet latencies are much smaller than in regular TCP. We get their mean and standard deviation of packet latencies shown in Figures 6 and 7. We can find that the mean and standard deviation of packet latencies in pFabric also increase with the increasing of workload and over-subscription. However, the mean and standard deviation of packet latencies are much smaller than in regular TCP and regular TCP with ELR, which means the difference of



(a) Latency with an oversubscription of $64:4:1$



(b) Latency with an oversubscription of $32:4:1$



(c) Latency with an oversubscription of $16:4:1$

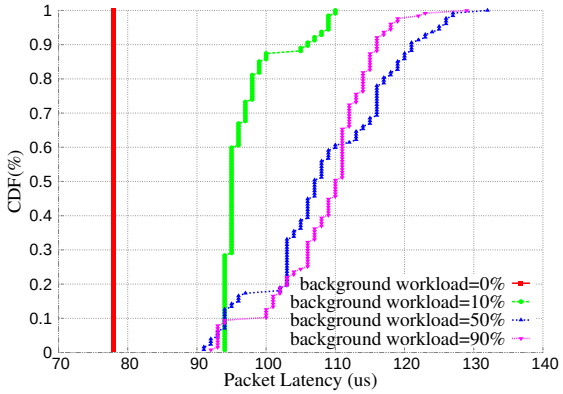
FIGURE 4. Packet latencies for pFabric

packet latencies in pFabric is much smaller than in regular TCP and regular TCP with ELR.

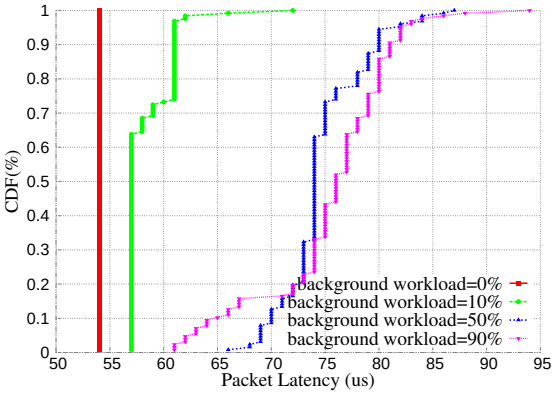
3.4. **Simulation for pFabric with equal-length routing.** As packet latency in pFabric is very small. We think the pathway length in pFabric may have more impact on the difference of packet latencies than in regular TCP. Here, we also use ELR in pFabric and Figure 5 shows the simulation results.

We can learn from Figure 5(a) that all packet latencies are the same as 78μs when there is no background workload. When we increase the workload, the packet latencies are also increased slightly. We get the similar results when the over-subscription is 32 : 4 : 1 and 16 : 4 : 1. This is because the background flows are created with diverse length of flows. So test packets may have the same priority as packets from 1-packet-short background flows and have to spend some time waiting for transmission when there are some short flow packets in switch queues. We can know from Figure 5 that ELR in pFabric cannot completely eliminate the difference of packet latencies yet. According to Figures 2, 3, 4 and 5, we find that packet latencies in pFabric with ELR are much smaller than that in Figures 2 and 3, but a little higher than that in Figure 4. The detailed mean and standard deviation of packet latencies in pFabric with ELR are also shown in Figures 6 and 7. We observe that the mean latencies in pFabric with ELR are only slightly increased than in pFabric; however, the standard deviations are obviously declined. It means that using ELR on pFabric can significantly minimize the difference of packet latencies in data center.

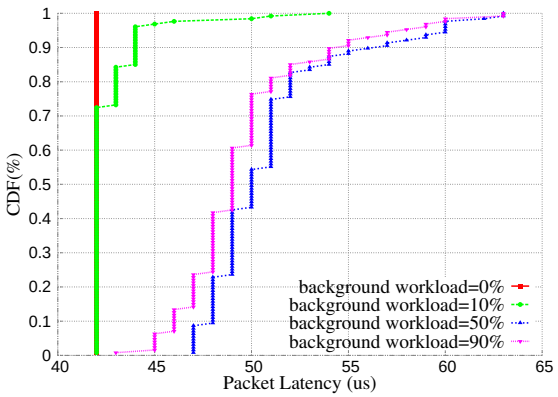
According to all subfigures in Figures 6 and 7, we can know that the background workload and over-subscription have a much higher effect on the difference of packet



(a) Latency with an oversubscription of 64 : 4 : 1

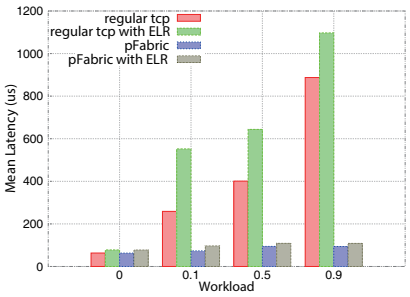


(b) Latency with an oversubscription of 32 : 4 : 1

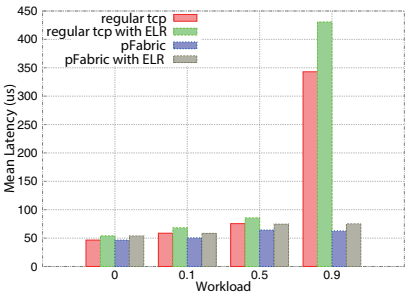


(c) Latency with an oversubscription of 16 : 4 : 1

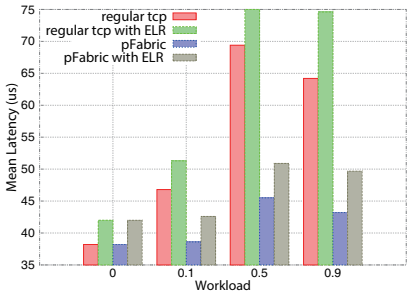
FIGURE 5. Packet latency for pFabric with equal-length routing



(a) Mean latency for oversubscription 64 : 4 : 1

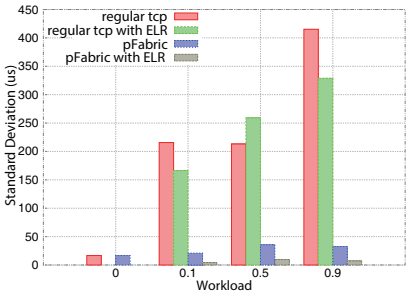


(b) Mean latency for oversubscription 32 : 4 : 1

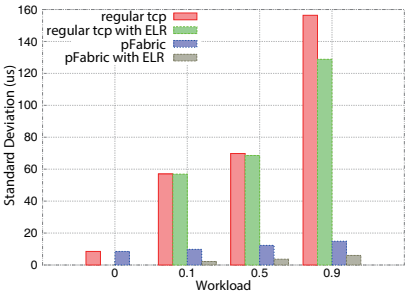


(c) Mean latency for oversubscription 16 : 4 : 1

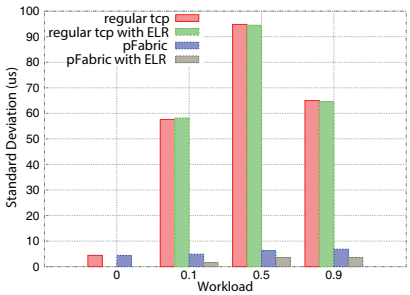
FIGURE 6. Mean latency



(a) STDDEV latency for oversubscription 64 : 4 : 1



(b) STDDEV latency for oversubscription 32 : 4 : 1



(c) STDDEV latency for oversubscription 16 : 4 : 1

FIGURE 7. STDDEV latency

latencies in regular TCP than in pFabric. When we use ELR both for regular TCP and pFabric, we observe that background workload and over-subscription still obviously affect the mean packet latencies on regular TCP, but have little impact on pFabric. With the help of ELR, the standard deviation of packet latencies significantly minimized in pFabric.

Based on these observations and analysis, we argue that with the help of ELR, pFabric shows an excellent performance in minimizing the difference of packet latencies in data center and can meet the requirement of most latency-sensitive systems. To the best of our knowledge, it is the first effort to integrate ELR into pFabric for minimizing the difference of packet latencies in data center.

4. **Conclusion.** In this paper, we do several simulations with different over-subscriptions and background workloads both for regular TCP and pFabric to conduct a study of the difference of packet latencies in data center. It is the first time to simulate pFabric with ELR in NS2 to test its performance in packet latency. Based on all the simulations and analysis, we find that there are two factors affecting the difference of packet latencies in data center: one is the different length of pathway for different packets; the other main factor is the background workload and over-subscription in data center. pFabric has a much better performance than other regular TCP protocols, and we think with the help of ELR, pFabric can meet the strict requirement of most latency-sensitive and real-time systems. As part of future work, we plan to do further research to achieve ELR in a real data center environment with pFabric.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (41301514), Key Laboratory of High Performance Computing of Jiangxi Province (PKLHPC1303), Jiangsu Science Foundation (BK20151245) and Huai'an City Science Foundation (HAG2015007).

REFERENCES

[1] S. Kandula, S. Sengupta, A. Greenberg, P. Patel and R. Chaiken, The nature of datacenter traffic: Measurements & analysis, *Proc. of the ACM SIGCOMM Internet Measurement Conference*, Chicago, IL, USA, pp.202-208, 2009.

[2] T. Benson, A. Akella and D. A. Maltz, Network traffic characteristics of data centers in the wild, *Proc. of the ACM Internet Measurement Conference*, Melbourne, VIC, Australia, pp.267-280, 2010.

[3] T. Benson, A. Anand, A. Akella and M. Zhang, Understanding data center traffic characteristics, *ACM SIGCOMM Computer Communication Review*, vol.40, no.1, pp.92-99, 2010.

[4] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta and M. Sridharan, Data Center TCP (DCTCP), *ACM SIGCOMM Computer Communication Review*, vol.40, no.4, pp.63-74, 2010.

[5] K. K. Ramakrishnan, S. Floyd and D. Black, *The Addition of Explicit Congestion Notification (ECN) to IP*, IETF, RFC 3168, 2001.

[6] M. Alizadeh, A. Kabbani, T. Edsall, B. Prabhakar, A. Vahdat and M. Yasuda, Less is more: Trading a little bandwidth for ultra-low latency in the data center, *Proc. of the 9th USENIX Symposium on Networked Systems Design and Implementation*, San Jose, CA, USA, pp.253-266, 2012.

[7] C. Y. Hong, M. Caesar and P. B. Godfrey, Finishing flows quickly with preemptive scheduling, *ACM SIGCOMM Computer Communication Review*, vol.42, no.4, pp.127-138, 2012.

[8] J. Cao, R. Xia, P. Yang, C. Guo, G. Lu, L. Yuan, Y. Zheng, H. Wu, Y. Xiong and D. Maltz, Per-packet load-balanced, low-latency routing for clos-based data center networks, *Proc. of the ACM International Conference on Emerging Networking Experiments and Technologies*, Santa Barbara, CA, USA, pp.49-60, 2013.

[9] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar and S. Shenker, pFabric: Minimal near-optimal datacenter transport, *ACM SIGCOMM Computer Communication Review*, vol.43, no.4, pp.435-446, 2013.

[10] *The Network Simulator NS-2*, http://www.isi.edu/nsnam/ns/.

[11] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel and S. Sengupta, VL2: A scalable and flexible data center network, *Communications of the ACM*, vol.54, no.3, pp.95-104, 2011.

[12] M. Fiorani, S. Aleksic and M. Casoni, Hybrid optical switching for data center networks, *Journal of Electrical and Computer Engineering*, vol.2014, 2014.

[13] D. Zats, T. Das, P. Mohan, D. Borthakur and R. Katz, DeTail: Reducing the flow completion time tail in datacenter networks, *ACM SIGCOMM Computer Communication Review*, vol.42, no.4, pp.139-150, 2012.