Journal of Network and Computer Applications 217 (2023) 103692

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Load balancing for heterogeneous traffic in datacenter networks ☆

Jin Wang [a], Shuying Rao [a], Ying Liu [a], Pradip Kumar Sharma [b], Jinbin Hu [a,*]

[a] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China
[b] Department of Computing Science, University of Aberdeen, Aberdeen, UK

ARTICLE INFO

Keywords:
Data center
Load balancing
Switching granularity
Multipath

ABSTRACT

In modern datacenter networks (DCNs), the overwhelming heterogeneous flows have various stringent demands, ranging from delay-sensitive short flows, throughput-sensitive long flows to best-effort flows without deadline. Recently, many load balancing schemes are proposed to deliver good performance for datacenter applications. However, the existing solutions cannot meet all the above requirements simultaneously. Especially, the short flows experience head-of-line blocking due to queued behind the long and best-effort flows. The long flows often suffer from throughput degradation due to bursty congestion. To solve these issues, we present LBT, a traffic-differentiated load balancer. The key design point of LBT is to adaptively adjust the switching granularity of long flows and best-effort flows according to the two calculated switching thresholds based on the traffic strength. Specifically, under heavy load, the switching granularity is increased to guarantee required bandwidth capacity for delay-sensitive short flows to finish quickly. In contrary, the switching granularity is reduced to enable the long flows to make full of parallel equal-cost paths. Moreover, we adopt different routing strategies for the three different categories flows. We conduct NS-2 simulations to evaluate the effectiveness of LBT. The experimental results show that LBT significantly reduces the average flow completion time of short flows by up to 55.9% ~ 65.4% compared to the state-of-the-art solutions and achieves high throughput for long flows concurrently.

1. Introduction

With the rapid development of the Internet of Things (IoT) and cloud computing technologies (Anon, 2020), datacenters as high-performance hardware infrastructure are becoming more popular and widely used (Luo et al., 2022; Jian et al., 2021). In order to build high availability, high performance and low-cost cloud computing infrastructure storage and computing facilities, datacenters usually deploy a large number of commercial switches and servers. The datacenter network connects large-scale server clusters and is a bridge for transferring computing and storing data. In general, the datacenter is ideally equipped to provide network services with high throughput and low latency. Managing the traffic in the datacenter network can improve the overall utilization of the network link, reduce the congestion of the network, and reduce the retransmission in the transmission process. Therefore, it is critical to design a reasonable and efficient load balancing scheme for the datacenter network among the available multiple paths.

In recent years, many load balancing schemes have been proposed to make full use of parallel paths in datacenters. Whether CONGA (Alizadeh et al., 2014) and LetFlow (Vanini et al., 2017) are transmitted at the flowlet granularity, or RPS (Dixit et al., 2013) and DRILL (Ghorbani et al., 2017) are transmitted at the packet level, there are inevitable defects in the transmission process. RPS and DRILL divide flows at packet granularity, and select the next hop for each packet according to the local queue length to make use of multiple paths. In this way, although they can make full use of link resources and improve link utilization, they also cause the problem that data packets are easy to be out of order. While CONGA and LetFlow reroute the flows with the flowlet granularity, which will not cause too many packet reordering. However, due to the mechanism properties of them, the feedback time of CONGA is prolonged and LetFlow adopts random routing. Disadvantages of CONGA and LetFlow are easy to cause link congestion, high delay and other shortcomings.

In addition, for flows smaller than 100 KB, Hermes (Zhang et al., 2017) uses routing without switching. Other flows use packet transmission granularity for multi-path forwarding, and make rerouting decisions according to path status and flow status. However, although Hermes distinguishes between traffic, it still ignores the precondition of short flows priority. When it is impossible to meet the requirements

☆ This work is supported by the National Natural Science Foundation of China (62102046,62072056), the Natural Science Foundation of Hunan Province, China (2022JJ30618, 2020JJ2029), and Scientific Research Fund of Hunan Provincial Education Department (22B0300).

* Correspondence to: Changsha University of Science and Technology, Changsha 410004, China
E-mail address: jinbinhu@csust.edu.cn (J. Hu).

https://doi.org/10.1016/j.jnca.2023.103692
Received 3 December 2022; Received in revised form 25 February 2023; Accepted 15 June 2023
Available online 1 July 2023

of short flows with low delay and long flows with high throughput at the same time, short flows should be given priority. That is, the first condition is to meet the deadline for short flows, and then the goal is to achieve high throughput for long flows.

At present, most existing load balancing schemes do not realize the problem of heterogeneous network traffic characteristics. About 80% of the traffic is only provided by about 20% of throughput sensitive long flows, and about 80% of delay sensitive short flows only provide about 20% of the traffic (Alizadeh et al., 2010; Munir et al., 2013; Benson et al., 2010). In addition, there is another type of flow in the datacenter called best-effort (BE) flow (Zhang et al., 2022; Chen et al., 2016), which does not require low latency and high throughput, such as background backup traffic and other mass storage tasks. Therefore, in the large environment of datacenter network, if all flows are rerouted with the same granularity, many adverse consequences will occur. Long and short flows will compete for link resources, short flows are prone to long tail queuing delays, as well as excessive use of link resources by the BE flows. In this way, average flow completion time (AFCT) of short flows will be prolonged and the throughput of long flows will be reduced.

In view of the above shortcomings, this paper proposes a load balancing mechanism called LBT. It divides the flows into short flows, long flows and BE flows according to the different traffic characteristics. On this basis, we set two thresholds in the transmission queue. These two thresholds change dynamically in real time, taking the load strength of long and short traffic and the load strength of the whole link as the impact conditions respectively. During transmission, we prioritize short flows and transmit short flows at flow level to avoid being out of order. The queue length chosen for short flows is the shortest queue on the links, which leaves sufficient paths for short flows and greatly reduces the probability of congestion. Both long flows and BE flows use adaptive granularity of transmission. Among them, long flows and BE flows select the port with the longest queue length within the first threshold and the second threshold respectively. This effectively avoids performance degradation caused by competition between heterogeneous flows, thus guaranteeing low latency for short flows and throughput for long flows. In addition, LBT only needs to be deployed on the switches.

The main contributions of this paper are as follows:

- We conduct in-depth research to analyze the two main issues brought on by the transmission of heterogeneous traffic at the same granularity: throughput and link utilization decline due to the disorder of the coexistence of long and short flows, and delay increases as a result of the long tail congestion of short flows.
- We propose a load balancing scheme LBT, which introduces BE flow and calculates different thresholds by sensing traffic load, so as to flexibly switch the transmission granularity of long flows and BE flows. In order to prevent long tail congestion and frequent disorder, LBT specifically determines the switching granularity of long flows and BE flows based on the load strengths of long flows and short flows as well as the link's overall load strength.
- We reroute different types of traffic in accordance with certain thresholds in order to achieve low latency and high throughput. By avoiding long flows multi-path transmission while the long flows and short flows are on the same path, this will give short flows greater link resources. The results of our theoretical analysis of threshold setting's efficacy demonstrate that doing so significantly lowers the probability of short flows being congested and increases the throughput of long flows.
- We illustrate that the performance of LBT is significantly superior than the traditional load balancing scheme using NS-2 to simulate the traffic scenarios of Web Server, Web Search, and Data Mining. In particular, LBT decreases the AFCT by 55.9% ~ 65.4% for short flows under 0.5 load in Data Mining scenario when compares to ECMP, CONGA, DRILL, LetFlow, and Hermes.

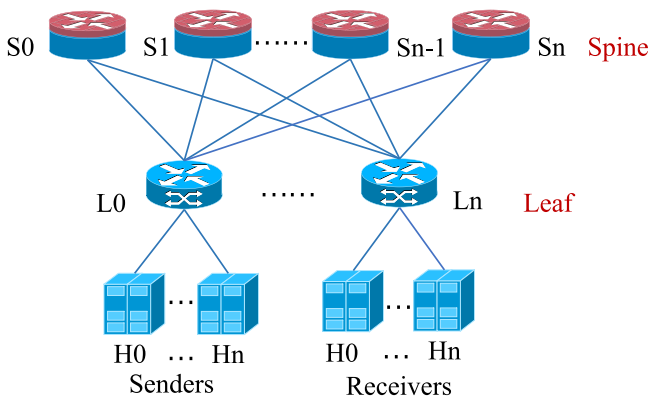


Fig. 1. Leaf-spine topology.

The rest of the paper is organized as following. We set up the motivation in Section 2. In Sections 3 and 4, we describe the overview of design and introduce the details of LBT, respectively. We discuss the implementation in Section 5. In Section 6, we show large-scale NS-2 simulation results. In Section 7, we present the related work and then conclude the paper in Section 8.

2. Motivation

2.1. Traffic differentiations are ignored

As we all know, the datacenter contains tens of thousands of flows (Wang et al., 2022; Hu et al., 2018; Wang et al., 2021), and the data flows produced by various applications require various levels of network transmission performance (Wilson et al., 2011; Vamanan et al., 2012). According to the traffic distribution in the datacenter, traffic is often classified into long flows and short flows using the 100 KB (Chen et al., 2016; Kheirkhah et al., 2016) measurement standard. Service applications with severe latency requirements, like online real-time search, information interaction, etc., are the main sources of short flows. High network throughput applications, such file backup, are the main source of long flows. There are, however, some datacenter flows that simply call for a limited quantity of transmission. These flows, collectively known as BE flows, do not need to respond quickly or produce output with a high throughput. Moreover, there are no task deadlines in this flow. When transferred inside a datacenter, they just need to be sent completely from the sender to the receiver. Hence one can see that the transmission priority of BE flows is obviously lower than that of long flows and short flows. It can automatically reduce the transmission volume when the link is busy, or even suspend the transmission to alleviate the link congestion. In addition, BE flows do not compete with long flows and short flows for link resources. This advantage can allow long flows and short flows to have more link resources, thus achieving the transmission effect of low latency and high throughput.

However, BE flows are not considered in the existing load balancing schemes (e.g. ECMP Zhang et al., 2014 and Hermes). Most schemes choose to ignore the flows characteristics so as not distinguish flows or simply distinguish flows. Simply divide flows into long flows or short flows according to the rule of whether the flow size is greater than 100 KB. In the scheme of these two classifications, ECMP does not distinguish between long flows and short flows. It uses Hash functions to disperse long flows and short flows to equivalent multipath. Although this ensures the fairness of the transmission, it will inevitably lead to the negative impact of long and short flows going the same path. Specifically, long flows blocking short flows, short flows waiting time is too long and easy to lead to link congestion. At the same time, although Hermes complies with the demand of traffic, it uses

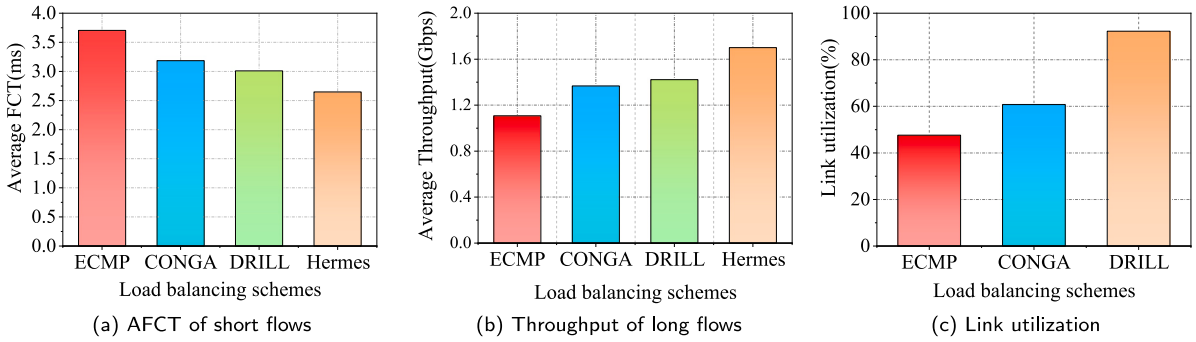


Fig. 2. Performance under different load balancing schemes.

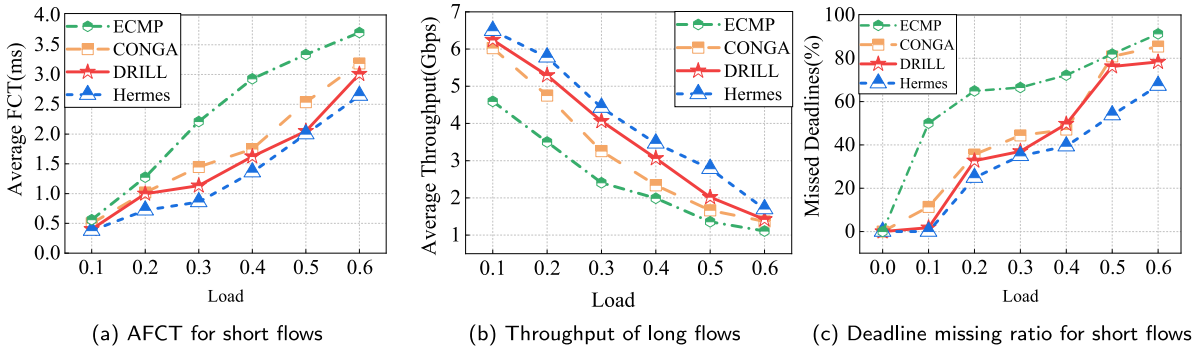


Fig. 3. Performance under different traffic loads.

the method of dividing traffic by 100 KB. However, it does not take the BE flows mentioned above into account, which may cause the BE flows to occupy too much link resources. In this way, the transmission performance of long and short flows will be poor. In a word, compared with not distinguishing traffic or dividing it into two types of traffic, the operation of dividing traffic into three types will be more detailed and more responsive to the transmission requirements of internal traffic in DCN.

2.2. Rerouting at the same granularity falls short

It can be seen from the traffic characteristics of the datacenter that the length of the data flow is heavy tailed. That is, 90% of the data flows come from short flows, while it only provides 20% of the data. At the same time, 10% of the data flows from the long flow provide 80% of the data (Hu et al., 2019b, 2021). It can be seen that there are great differences in the frequency and size of long and short flows. If all flows are transmitted at the same granularity, it will inevitably waste link resources, cause link congestion, and have other undesirable consequences.

The traffic of the datacenter is ever-changing. In order to meet the transmission requirements of low latency and high throughput, many typical load balancing schemes emerge at the historic moment. In the current classic load balancing schemes. There are ECMP with flow as the transmission unit, RPS and DRILL with packet granularity, CONGA and LetFlow with flowlet granularity. Also, Hermes that separate traffic and transmit them separately. In order to compare the advantages and disadvantages of various load balancing schemes, we used NS-2 simulation to carry out a series of experiments. We use leaf-spine topology in the experiment, as indicated in Fig. 1. The buffer size of each switch is 256 packets, the bandwidth of each path is 10 Gbps, and the round-trip propagation latency is 100 μs.

The performance of DCN transmission often depends on the transmission of long flows and short flows. Therefore, we compare the AFCT of short flows and the average throughput of long flows in various schemes. It can be seen from (a) and (b) of Fig. 2 that the AFCT of short flows of ECMP is significantly higher than that of DRILL and Hermes. This is because ECMP adopts a transmission mode that does not distinguish between long and short flows. Short flows are prone to long tail delay, which leads to an increase in AFCT. DRILL adopts packet granularity transmission, which is easy to make short flows retransmit out of order, and ultimately lead to the increase of short flow delay. Meanwhile, when comparing the average throughput of long traffic, we can find that the flow granularity transmission scheme has lower load than other granularity transmission schemes, and Hermes has the highest performance due to its ability to distinguish traffic. ECMP uses hash hashing to schedule flows, and different flows may choose the same fixed path for forwarding, which can easily cause network congestion and reduce throughput. For Hermès, although its performance is better than that of other schemes, due to its global congestion awareness and conservative routing characteristics, it cannot achieve optimal routing in the transmission process. It can be seen that the scheme of transmission with the same granularity is difficult to achieve balance between short flows requiring low latency and long flows requiring high throughput.

In addition, link resources are extremely valuable. During network transmission, we should try our best to avoid link waste and ensure the highest link utilization. In order to compare which granularity has the highest throughput from the perspective of transmission granularity, we select load balancing schemes representing different granularity transmission for comparison. ECMP stands for flow, Conga stands for flowlet, and DRILL stands for package granularity scheme. From our experimental results, we can see that the link utilization of packet granularity DRILL is the highest, followed by the flowlet granularity CONGA. And the last one is ECMP, which is prone to link congestion and is transmitted at flow granularity. It can be seen that packet granularity transmission is the most friendly indicator of link utilization, while flow granularity transmission will greatly reduce the utilization of link resources. Refer to Fig. 2(c) for details.

In order to further judge the advantages and disadvantages of various schemes under different conditions, we conduct experiments

on mixed traffic under different load intensities. In the experiment, the mixed traffic transmission mode is adopted, that is, the long and short flows are transmitted at the same time. The load intensity represents the density of the long and short flows on the link, the value range is 0 to 1. The closer the value is to 1, the greater the load intensity. It can be seen in Fig. 3(a) that when the load increases, the AFCT of short flows will increase. The changes in Hermes and ECMP are obvious, because Hermes does not prioritize short flows, and ECMP is short flows typically experience long tail congestion. On the other hand, the average throughput of long flows in all schemes rapidly decreases as the load is increased. Even when the load strength reaches 0.6, the average throughput is only one sixth of that when the load strength is 0.1. For details, refer to Fig. 3(b).

Finally, there are often many short flows with deadline in DCN. To ensure the quality of service, these short flows must be completed within the deadline. In order to compare the transmission performance of various load balancing schemes for this type of traffic, we define the deadline for each short flows to be consistent with other articles (Alizadeh et al., 2014; Vanini et al., 2017; Dixit et al., 2013; Ghorbani et al., 2017; Zhang et al., 2017). It can be seen from Fig. 3 that with the increase of load strength, the performance of all load balancing schemes decreases. Moreover, the disadvantages of ECMP performance are very obvious. The probability of short flows missing the deadline can be as high as about 90%. Hermes, which takes a long time to perceive global congestion routing, and DRILL, which is prone to packet chaos, still have a high probability of missing the deadline.

After comparing several indicators such as the AFCT of short flows, the average throughput of long flows and link utilization, we can easily find that the existing load balancing schemes for the same granularity transmission are difficult to balance each other. In packet granularity transmission, there is often an imbalance between high link utilization and high latency. At flow level, the contradiction between high throughput of long flows and high delay of short flows is more intense. Even CONGA based on flowlet transmission cannot obtain optimal solutions to these problems. Therefore, the transmission mode with the same granularity in DCN cannot meet all kinds of transmission requirements, and many defects cannot be avoided.

3. Design overview

In this section, we will provide a detailed overview of LBT. The key to LBT is to divide DCN traffic into three different types of traffic and adaptively adjust the switching granularity for long flows and BE flows. This allows to choose the best way for each packet per flow, reducing queuing time for short flows and ensuring multi-path transmission for long flows. To be more specific, short flows are transmitted at the granularity of flow and are routed in the shortest queue to avoid packet retransmission and link congestion. On the other hand, we calculate two different thresholds according to different load intensities, so that long flows and BE flows can be adjusted to select routes according to the threshold while switching granularity, so as to achieve short flows priority and long flows multi-path transmission. LBT consists of three modules, as shown in Fig. 4.

(1) **Traffic differentiation model:** In LBT, we divide the traffic transmitted within the DCN into three types of traffic according to their different transmission needs, namely short flows, long flows, and BE flows. Among them, the premise is to reduce the time delay of short flows (Hannabuss, 2012) and improve the transmission performance of the long flows at the same time.

(2) **Threshold calculation model:** The paper of threshold calculation is carried out in the switches, which mainly includes load intensity estimation (long and short flows load and overall link load) and threshold calculation. To be more specific, the ratio of the link's existing long and short flows to its maximum transmission capacity determines the first load intensity in real time. Then, determine the switching granularity of the long flows, which is the first threshold. The ratio of all existing flows on the link to the maximum transmission of the whole links is used to calculate another load intensity. Finally, the switching granularity of BE flows is computed as the second threshold.

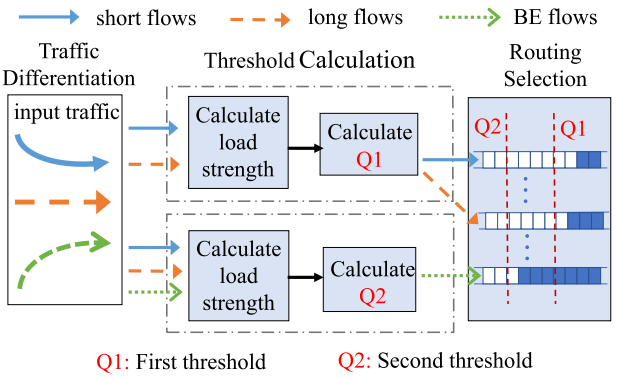


Fig. 4. LBT architecture.

(3) **Routing selection model:** In addition to selecting the forwarding path based on the output port's real-time queue length, the routing model is primarily in charge of switching long flows and BE flows with different granularities in accordance with the threshold. Due mainly to the requirements for giving short flows priority protection, short flows use the link's shortest queue port to minimize congestion and shorten queue times. Long flows choose the longest queue port available inside the first threshold at the same time in an effort to minimize long flow congestion and accomplish multi-path transmission. Finally, BE flows select longest lost port within the second threshold in an effort to avoid competing for resources with long and short flows.

4. Design details

4.1. Threshold calculation

In DCNs, there are several long flows with high throughput to maintain and numerous short flows with extremely short AFCT to ensure. As a result, the objective of this paper's research is to develop a load-balancing system that satisfies the requirements of diverse traffic transmissions. In our design, we introduce BE flows based on the fundamental ideas of long and short flows, and we regulate the switching granularity of both based on various thresholds. In this way, BE flows will not use up too many link resources when the low delay transmission requirements of short flows are met, enabling the multi-resource transmission of long flows. We employ the queuing model for threshold calculations in order to meet the design goals.

As we all know, the short flows in DCN are very bursty (Hu et al., 2019a), in which the flow size and number are ever-changing. In accordance with this feature, LBT will determine the long and short traffic loads as well as all traffic loads in real-time, and it will periodically update the threshold to effectively prevent the negative effects brought on by the fixed threshold's inapplicability to different transmission conditions. Because prior research (Alizadeh et al., 2014) has shown that the inactive interval between the two bursts is 500 μs, LBT simultaneously adjusts the time interval for dynamically updating the two thresholds to this value.

Theoretically, when the traffic load increases, the threshold value will increase accordingly. And the handover granularity of long flows and BE flows will be increased to guarantee that there are sufficient transmission paths for short flows, and avoid congestion of short flows resulting in long queuing delays. At the same time, when the traffic load decreases, the threshold will decrease. After the handover granularity of long flows and BE flows is reduced, the long flows and BE flows can be flexibly switched on multiple paths to improve the link utilization

Table 1
Key parameters in LBT.

Symbol	Represents meaning
K	The number of all equivalent paths
R	Round-trip propagation delay
W	Maximum window size (64 KB)
l	Load strength
G	Bottleneck link capacity
Ps Pl Pb	Number of short flows, long flows and BE flows
Xs Xl Xb	Average byte size of short flows, long flows and BE flows
Ks Kl Kb	Number of links of short flows, long flows and BE flows

and long flows throughput. All in all, the adaptive granularity of long flows and BE flows can ensure the transmission requirements of both latency-sensitive short flows and throughput-sensitive long flows.

Because queueing models with generic interarrival time distributions are difficult to evaluate and there are few findings (Hu et al., 2019a), we construct our model on partial assumptions. We take into account the phenomenon of traffic transmission timeouts and retransmissions in DCNs, but abstract the transmission link of DCN as M/G/1 FCFS queueing model with unlimited buffers. To get a clear picture of the model-building process, we show some key symbols in Table 1.

First, LBT calculates the first load strength based on the ratio of the existing long flows and short flows load to the link capacity l_1.

$$l1 = \frac{P_s \cdot X_s + P_l \cdot X_l}{K \cdot G} \tag{1}$$

Then, another load strength is estimated according to the ratio of all existing flows on the links to the maximum transmission of the entire link l_2.

$$l2 = \frac{P_s \cdot X_s + P_l \cdot X_l + P_b \cdot X_b}{K \cdot G} \tag{2}$$

In the process of transmission, the long flows will be transmitted to the longest queue port within the first threshold by default, and the BE flows will be transmitted to the longest queue port within the second threshold. When the queue length of the long flows being transmitted is greater than Q_1 (The values of Q_1 and Q_2 are updated in real time), the long flows will be rerouted to the next longest queue port within the first threshold. Long flows of packets transmitted on the link ($Q1 \cdot K_l$)and the number of packets queued on the link($K_l \cdot t \cdot G$) is equal to the total amount of data in the long flows.

Like the long flows, when the queue length of the BE flows being transmitted is greater than Q_2, the BE flows will also be rerouted, and it will be rerouted to the next longest queue port within the second threshold. Long flows of packets transmitted on the link ($Q2 \cdot K_b$)and the number of packets queued on the link($K_b \cdot t \cdot G$) is equal to the total amount of data in the long flows.

Because short flows are fewer than 100 KB in DCN, their sizes are pretty small (Xu and Li, 2014; He et al., 2022). As a result, the transmission of these short flows is finished during the sluggish start period. Sluggish start period means that every time the TCP receiving window receives an acknowledgment, it will grow, that is, each short flow will first send $2m$ packets, then $4m$, $8m$, etc. Therefore, the number of RTT rounds required to complete the short flow transmission of X_s bytes is $\log_2 \frac{X_s}{m} + 1$.

During transmission, short flows are transmitted with the flow as the transmission unit, so the FCT of the short flows: $FCT_s = \sum_{q=1}^{n} \frac{W_q}{X_s} + \frac{X_s}{G}$, and the $\frac{X_s}{G}$ is the transmission delay. Then the FCT formula of short flows is as follows.

$$E[FCT_s] = (\log_2 \frac{X_s}{m} + 1) \cdot \frac{E[W_q]}{X_s} + \frac{X_s}{G} \tag{3}$$

The M/G/1-FCFS queuing model is used to simulate the average queuing delay (Chen et al., 2016; Xu and Li, 2014; Alizadeh et al., 2013), and $E[Wq]$ may be estimated using the well-known Pollaczek Khintchine method.

$$E[W_q] = \frac{l}{1-l} \cdot \frac{1+C_S^2}{2} \cdot E[S] = \frac{l}{1-l} \cdot \frac{E[S]^2}{2E[S]} \tag{4}$$

where C_S^2 is the squared coefficient of variation of the service distribution, and $C_S^2 = \frac{Var[S]}{E[S]^2}$. In addition, S represents the service time, E[S] represents the average service time of each short flow, that is $\frac{W}{G}$. So, $E[W_q]$ can be calculated as.

$$E[W_q] = \frac{l}{2(1-l)} \cdot \frac{W}{G} \tag{5}$$

Thus, according to the above Eqs. (1), (4) and (5), we can get the FCT of short flows formula related to the first threshold as follows.

$$E[FCT_s] = \frac{W(\log_2 \frac{X_s}{m} + 1)(P_s X_s + P_l X_l)}{2GX_s((K_s + \frac{P_l X_l}{(Q1+tG)} + K_b)G - P_s X_s - P_l X_l)} + \frac{X_s}{G} \tag{6}$$

And, according to the above Eqs. (2), (4) and (5), we can get the AFCT of short flows formula related to the second threshold as follows.

$$E[FCT_s] = \frac{W(\log_2 \frac{X_s}{m} + 1)(P_s X_s + P_l X_l + P_b X_b)}{2GX_s((K_s + K_l + \frac{P_b X_b}{(Q2+tG)})G - P_s X_s - P_l X_l - P_b X_b)} + \frac{X_s}{G} \tag{7}$$

Finally, we get the first thresholds as

$$Q1 = \frac{P_l X_l}{Z1 - K_s - K_b} - Gt \tag{8}$$

where $Z1$ is

$$Z1 = \frac{W(\log_2 \frac{X_s}{m} + 1)(P_s X_s + P_l X_l)}{2(E[FCT_s] - \frac{X_s}{G})X_s G^2} + \frac{P_s X_s + P_l X_l}{G} \tag{9}$$

The expression of the second thresholds as

$$Q2 = \frac{P_b X_b}{Z2 - K_s - K_l} - Gt \tag{10}$$

where $Z2$ is

$$Z2 = \frac{W(\log_2 \frac{X_s}{m} + 1)(P_s X_s + P_l X_l + P_b X_b)}{2(E[FCT_s] - \frac{X_s}{G})X_s G^2} + \frac{P_s X_s + P_l X_l + P_b X_b}{G} \tag{11}$$

Q_1 and Q_2 are the two thresholds on the links. Specifically, the switch selects routes according to the threshold size and the real-time queue length on the link. If the queue length on the link exceeds Q_1 at this time, the long flows will select another link with the longest queue length within Q_1 for rerouting. In the same way, BE flows are judged according to the value of Q_2. In addition, according to Formulas (8) and (10), the values of Q_1 and Q_2 are closely related to the flow completion time of short flows. It can be seen that the two thresholds obtained by sensing different traffic loads including short flows are elastic to traffic patterns with time changes.

4.2. Routing selection

To reduce the negative impact of interaction between the various flows, LBT uses two different thresholds to toggle granularity boundaries and uses different strategies to transmit the three flows. As shown in Algorithm 1, three flows have different routing selections:

For short flows, they have low latency transmission requirements. When routing, in order to minimize waiting time, short flows will always choose the queue on the link with the least queue length. In addition, in LBT, the transmission will be based on the transmission granularity of the short flows to avoid the transmission delay caused by disorder.

For long flows, their priority is lower than that of short flows, so their path selection is the port with the longest queue length within the first threshold. They take the first threshold as the boundary of

handover granularity. When a new message arrives at the switch and the queue length on the link reaches the first threshold, it will select the handover transmission path and reroute the message to other ports with the longest queue length within the first threshold. In this way, the high delay caused by long and short flows in the same path and short flows waiting for long flows transmission can be reduced, and the low delay transmission of short flows can be guaranteed. In extreme cases, the existing queue length has reached the first threshold. At this time, the long flows packets are forwarded to the port with the longest queue on the link, and the short flows transmission still takes priority.

For BE flows, their priority is the lowest, and their path selection is the port with the longest queue length within the second threshold. Their path selection is similar to that of long flows, but their switching granularity is based on the second threshold. According to the threshold calculation, the greater the load, the greater the threshold. Compared with the first threshold, the second threshold increases the load of BE flows. Therefore, the second threshold will always be greater than the first threshold. When the load is heavy, it can be transmitted on one link without affecting other links. On the contrary, when the load is small, the handover granularity will be larger than that of the long flows, so that the long flows has more link resources. Even when the link is congested, the BE flows can be paused to free up more link resources. It is worth noting that the link load will be different at each time, and the two thresholds will change continuously in each t time period (t is 500 μs by default).

Algorithm 1: Rerouting Algorithm

Input:
The first threshold $Q1$;
The Second threshold $Q2$;
Queue length queued on the link $Qlength$;
Output:
Port number of the optimal path P_{best};
for *per packet* **do**
/*Rerouting for short flows*/;
Explore the shortest queue $Min(Qlength)$;
$P_{best} = P_{Min(Qlength)}$;
/*Rerouting for long flows*/;
if $Qlength < Q1$ **then**
Mark $Qlength$ as $Qlength1$;
Explore the longest queue in Q1 range $Max(Qlength1)$;
$P_{best} = P_{Max(Qlength1)}$;
else
Explore the longest queue $Max(Qlength)$;
$P_{best} = P_{Max(Qlength)}$;
end
/*Rerouting for BE flows*/;
if $Qlength < Q2$ **then**
Mark $Qlength$ as $Qlength2$;
Explore the longest queue in Q2 range $Max(Qlength2)$;
$P_{best} = P_{Max(Qlength2)}$;
else
Explore the longest queue $Max(Qlength)$;
$P_{best} = P_{Max(Qlength)}$;
end
return P_{best}
end

5. Implementation

To implement the LBT, we fully understand the fundamentals of various traffic and transmission requirements in the datacenter. The first is to lessen the AFCT of the short flow and prevent the detrimental effects of the long tail obstruction brought on by long and short flows traveling in the same direction. The second is to employ multi-path transmission to increase link resources and enhance the transmission of long flows that are throughput-sensitive. Furthermore, we point out that BE fluxes, another sort of background flow, do not demand fast throughput or low latency. In the event of link load, it can coordinate efforts to reduce link congestion.

Commercial programmable switches can be used, and the P4 programming language can be used to specify how the switch handles packets because of the design features of LBT area diversion, threshold calculation, and multiple routing. The BE flows will automatically be recognized from the other two flows due to a priori knowledge. When dealing with long and short flows, the host first interprets them as short flows. If the sent field size indicated on the host side is greater than 100 KB, it is divided into long flows. The P4 switch can supply registers and grow the number of registers by employing more static random access memory (SRAM) to expand storage since LBT will update the threshold frequently, which contains numerous variables (such as queue length, traffic quantity, traffic size, and other factors). In this way, the P4 switch provides an opening for the LBT to actually land while also offering hardware support for bettering the transmission efficiency of both long and short flows.

We prioritize short flows above all others in terms of transmission granularity and routing in order to meet their minimal latency transmission requirements. To prevent disordering and retransmission, we carry them at the flows granularity first. To ensure that brief flows always select the port with the shortest backlog on the link, we fully guarantee their transmission during routing. We additionally choose the longest queue in addition to using thresholds to regulate the switching granularity of the other two flows in order to further lower the AFCT of short flows. By doing this, large delays brought on by lengthy queuing of brief flows can be efficiently avoided. In this approach, under specific circumstances, short flows can still be transmitted normally despite the existence of a traffic burst.

Additionally, we determine the threshold by detecting the link load in the switch, and then we use the threshold to control the switching granularity between long flows and BE flows. In particular, the long flows are transmitted packet by packet, with each packet's routing decided by the threshold and the length of the current queue. The queuing packets will choose the longest queue with the following queue length falling inside the first threshold for routing and forwarding once the queue length hits the threshold. We set $Q2$ to be bigger than $Q1$, meaning that the switching granularity of BE flows is greater than that of long flows, in order to give the long flows the transmission advantage. When the load is heavy, BE flows are concentrated more on a single link in this fashion. Long flows will have more transmission pathways when the load is not heavy.

6. Evaluation and discussion

We adopt three traditional application scenarios – Web Server, Web Search, and Data Mining – to test the performance of LBT in large-scale settings. In terms of traffic distribution, the Web Server has a long flow to short flow ratio of roughly 1:4 and a short flow size of only 1M. In the hypothetical Web Search situation, more than 95% of the bytes are provided by 30% of the flows larger than 1 MB. About 3.6% of the flows greater than 35 MB in the Data Mining scenario provide 95% of the bytes, whilst approximately 80% of the flows are smaller than 100 KB, resulting in a heavy tail distribution (Zhang et al., 2019; Zhou et al., 2018). Web Search just indexes data and does not require additional storage space, whereas Web Server runs on the server and occupies more space in terms of spatial distribution. Also, data from Data Mining is more unpredictable and unreliable, requiring more space.

We compare LBT with five load balancing schemes: ECMP (Zhang et al., 2014), CONGA (Alizadeh et al., 2014), DRILL (Ghorbani et al., 2017), Hermes (Zhang et al., 2017) and LetFlow (Vanini et al., 2017). The data packet is distributed to the equivalent multi-path using static

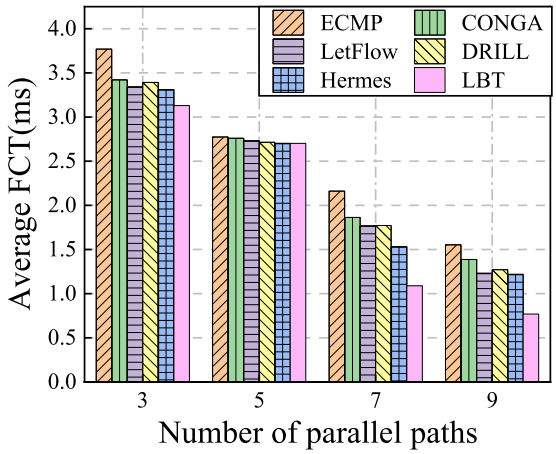


Fig. 5. AFCT for short flows with increasing number of paths.

hash in ECMP, which employs the flow as the transmission unit. With the flowlet serving as the transmission unit for CONGA and Letflow, CONGA chooses the best next hop for the traffic based on real-time feedback of the global information of the end-to-end path condition. Letflow classifies packet clusters based on predetermined intervals of time and randomly chooses forwarding ports for each cluster. Drill chooses the least loaded port as the packet forwarding port by comparing the queue lengths of the two currently randomly chosen ports with the least loaded port from the past. Drill uses the packet as the transmission unit. Hermes, which divides traffic transmission, decides whether to reroute short flows at the flow level or long flows at the packet level depending on the status of the path and the flow.

We utilize the leaf-spine topology, which has 8 leaves and 9 spines, as the transmission architecture in line with the motivation experiment. The whole network's switch buffer capacity is 256 packets, and each host is connected through a 10 Gbps link with a 100 μs round-trip propagation delay. Moreover, all flows are created via the Poisson process between random host pairings. To fully assess the performance of LBT, the standard DCTCP (Alizadeh et al., 2010) transmission protocol will also be employed, and the load will range from 0.1 to 0.7. It is worth noting that the two thresholds will be 500 μs is periodically updated, which means that the switching granularity between long flows and BE flows will change periodically.

6.1. Performance under Web Server workload

6.1.1. Performance with varying paths

In DCNs, many-to-many transmissions of data are often sent and transmitted. Therefore, a decent load balancing scheme must be compatible with multi-path transmission environments. In this section, we run simulation tests with various path numbers to see if LBT works well in a multi-path network setting. As the number of routes increased, we chose four transmission scenarios with 3, 5, 7 and 9 links to examine the effects of various load schemes on AFCT of short flows. In Fig. 5, the experimental results are displayed.

Theoretically, the greater the number of transmission links, the greater the number of paths from which to pick, and the effect of short flows will similarly trend lower. As shown in Fig. 5, the AFCT of short flows in each of the six load balancing schemes reduces as the number of routes rises. However, it is not difficult to discover that the AFCT of short flow differs when the number of links is changed and alternative load schemes are used. When there are 3 or 5 connections, the six load balancing strategies do not provide significantly different AFCT of short flows. However, it is worth noting that when there are 7 links, the AFCT of short flows in LBT is about 1 ms, which is only half of other schemes. Even when the number of links increases to 9, only the AFCT of short flows in LBT is less than 1 ms, and other schemes' AFCT is greater than 1 ms. This is because LBT adopts the transmission mechanism of multiple protection against short flows. The long flows choose the link with the longest queue length inside the first threshold whereas the short flows choose the link with the least link queue length on the existing path. It seeks to minimize the AFCT of short flows by preventing the long flows from obstructing the short flows and ensuring that the short flows are transported via the least obstructed links. However, Hermes relies on global congestion rerouting and has a long feedback time, which leads to a decline in the transmission performance of short flows. In addition, DRILL, LetFlow and CONGA of packet granularity and flowlet granularity transmission will also aggravate the occurrence of disorder due to the increase in the number of links. The short board of long flows obstructing short flows must be taken into account by ECMP, and the increase in the number of pathways will not make this fault go away. To sum up, LBT still performs well in terms of transmission despite the change in the number of links.

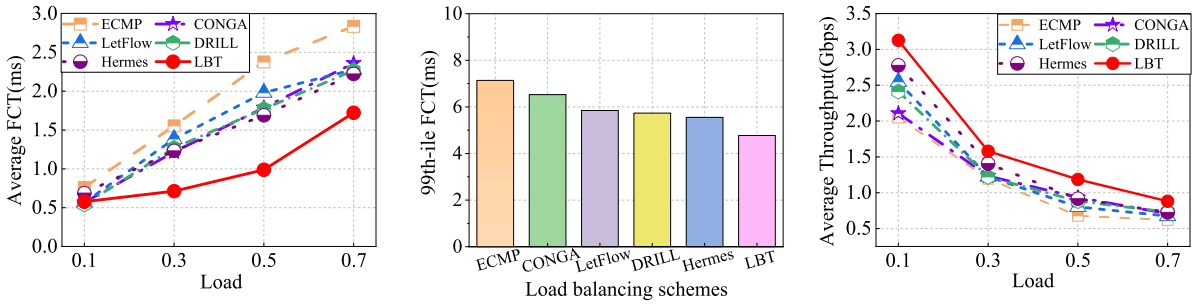
6.1.2. Performance with varying load

To further verify the performance of LBT, we compare it with five typical load balancing schemes under different load levels. In actual scenarios, some burst traffic will occasionally occur. Undoubtedly, they will increase the burden of the network, which is also an emergency that must be considered. Therefore, in order to determine if these load balancing strategies can still work well under high load strength, we compare them to the AFCT of short flows under various load intensities.

Fig. 6 (a) shows the AFCT of short flows with these load balancing schemes when the load ranges from 0.1 to 0.7. On the whole, with the increase of load, the AFCT of short flows in all schemes shows an upward trend. The result of this experiment accords with common sense. The heavy load in the DCN means that the traffic transmission competition is more intense at the same time, which is prone to congestion and packet loss. In detail, the AFCT of short flows in the four load balancing schemes, DRILL, CONGA, ECMP and LetFlow, have little difference with the increase of workload. On the contrary, the AFCT of LBT and Hermes under different workloads show obvious differences. Among them, when load reaches 0.5, the AFCT for short flows of Hermes increases rapidly. The possible cause is that Hermes uses global congestion awareness, which will generate an additional feedback delay. With the increase of the workload, the effect of the feedback delay is gradually aggravated. In addition, the LBT at the bottom is the most noteworthy. With the increasing workload, the AFCT of its short flows increases slowly, and the overall upward trend is relatively slow.

In order to analyze this result more clearly, we select the case of 0.3 workload for further analysis, and the result is shown in Fig. 6(b). As previously analyzed, the 99th-ile FCT in LBT is the smallest, and the results of the other five load balancing schemes are similar. Therefore, from the experimental results, LBT significantly reduces the flow completion time compared with five typical load balancing schemes. At the same time, as the workload increases, LBT also maintains more efficient performance.

In addition, long flows are not sensitive to delay requirements, but pay more attention to the throughput of the links during transmission. Therefore, we compare the throughput of LBT with five load balancing schemes under different loads. As shown in the experimental results in Fig. 6(c), as the workload increases, the throughput of long flows in all schemes decreases gradually. However, it can be seen from the figure that LBT's long flows throughput is always higher than other load balancing schemes. This is because the its long flows select the longest queue route within the first threshold, but the long flows select an adaptive granularity transmission mode, which can greatly increase the link utilization and thus the throughput of the long flows. In addition, we distinguish BE flows, which can attain more link resources for long flows.



(a) AFCT for short flows (b) 99th-ile FCT (c) Throughput for long flows

Fig. 6. Compare the performance in Web Server workload.

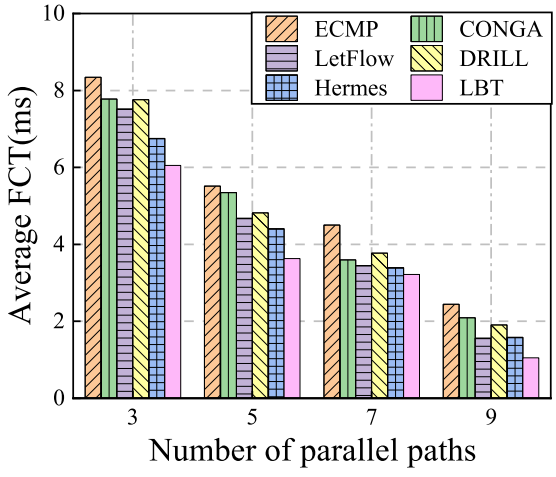


Fig. 7. AFCT for short flows with increasing number of paths.

6.2. Performance under Web Search workload

6.2.1. Performance with varying paths

In the larger scale Web Search scenario, we also simulate LBT and other load balancing schemes under different path numbers. In the experimental design, we continue to choose four transmission scenarios with path counts of 3, 5, 7 and 9, and we observe how the AFCT of short flows varies as the path count rises for these six load balancing schemes in the Web Search scenario.

Fig. 7 displays the AFCT for short flows in six load-balancing schemes with various numbers of paths. Roughly speaking, the AFCT of short flows steadily reduces as there are more transmission links available. When the number of links of LBT in the Web Search scenario is 3, 5, or 7, the AFCT of short flows is longer than it is in the Web Server scenario. This is because there are fewer links available to accommodate the demands of all traffic due to the increased scope and volume of the Web Search scenario. The AFCT of the entire short flows will rise if a portion of it is obstructed. To achieve effective transmission on the network, there must be enough pathways provided. In addition, the AFCT of short flows in LBT is lower than that of other schemes, which is consistent with the findings in the Web Server scenario. The AFCT of short flows in CONGA, DRILL, and ECMP is more than double that of LBT, even when there are 9 paths. Therefore, compared with other schemes, with the increase of the number of links, LBT is better for short flows transmission.

6.2.2. Performance with varying load

We continue to contrast LBT with five common load balancing schemes at various load levels in order to determine whether it can function in a situation at a wider scale. The AFCT of short flows with six schemes with various loads is shown in Fig. 8(a). It demonstrates that when the load increases, the AFCT of short flow in all schemes shows an upward trend. When the load is 0.1, the AFCT of short flows has little difference due to the small amount of flow. With the increasing workload, the AFCT of DRILL, CONGA, ECMP, LetFlow, and Hermes' short flows doubles. On the contrary, the increase of load has no significant impact on the performance of LBT, and its AFCT of short flows only slightly increases. In addition, in order to further explain the transmission performance of LBT. We adjust the load to 0.3 and test the 99th-ile FCT with multiple load schemes. As shown in Fig. 8(b), the 99th-ile FCT of LBT is reduced up to 50% compared with ECMP under 0.3 load. Therefore, it can be concluded that with the increase of workload, LBT also maintains more efficient performance.

Consistent with the Web Server, we compare the throughput of the long flows between LBT and five load balancing schemes under different loads. We also make experimental comparisons on workloads from 0.1 to 0.7, and the experimental results are shown in Fig. 8(c). As expected, the throughput of long flows will gradually decrease as the workload increases. LBT introduces BE flows and adopts adaptive granularity transmission mode, so it has higher throughput than other schemes.

6.3. Performance under Data Mining workload

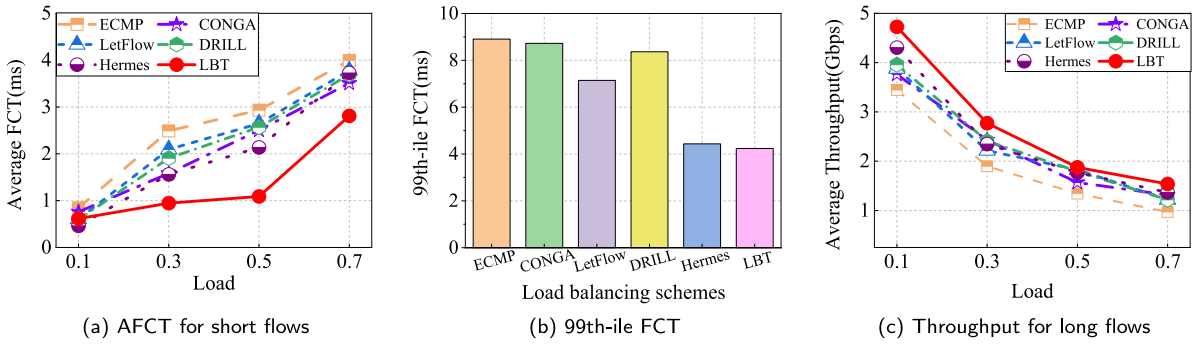
6.3.1. Performance with varying paths

We also simulate LBT and other load balancing schemes in the large-scale Data Mining scenario with varying numbers of paths. We keep the same number of paths in the experimental design to analyze how the AFCT of the short flows changed as the number of paths increased under the six load balancing schemes in the Data Mining scenario.

According to Fig. 9, in the Data Mining scenario, the AFCT of short flows steadily reduces as the transmission path increases. Also, the AFCT of short flows is higher than that of other scenarios because the traffic proportion and quantity of Data Mining are higher than those of Web Server and Web Search. In addition, regardless of the number of paths, the AFCT of short flows in LBT is lower than that of other schemes, which is consistent with previous scenarios. In light of this, LBT outperforms other schemes in terms of transmission performance by implementing multi-path transmission in a variety of large-scale scenarios.

6.3.2. Performance with varying load

We still compare with five typical load balancing schemes under various loads to determine whether LBT can apply to the transmission environment of Data Mining. The AFCT of short flows in all mechanisms exhibits a growing trend with increasing load, as illustrated in Fig. 10(a). The AFCT of short flows of DRILL, CONGA, ECMP, LetFlow, and Hermes grows exponentially as the load increases. On the other hand, the performance of LBT is not significantly impacted by the increase in load, and its AFCT of short flows only slightly increases. The AFCT of short flows of LBT is only one third of those schemes,



(a) AFCT for short flows (b) 99th-ile FCT (c) Throughput for long flows

Fig. 8. Compare the performance in Web Search workload.

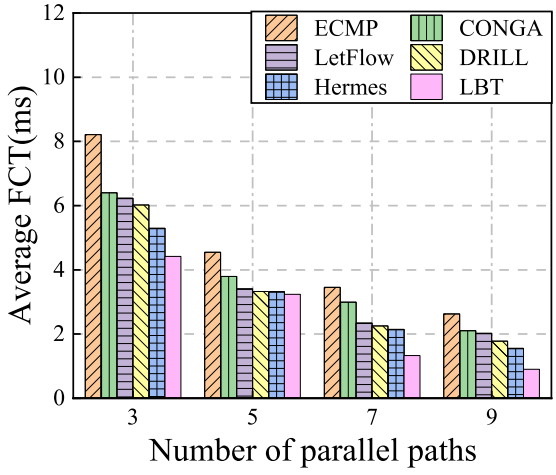


Fig. 9. AFCT for short flows with increasing number of paths.

even when the load is 0.5. In addition, it can be seen from 10 (b) that when the load is 0.3, the 99th-ile FCT of LBT is always smaller than that of other mechanisms, while ECMP is the highest due to its susceptibility to long tail blockage. As a result, it can be said that LBT has high transmission performance and can adapt to the transmission environment utilizing Data Mining.

In accordance with previous scenarios, we evaluate the throughput of long flows with LBT and five load balancing schemes under various loads. We still use different loads for the experiment, and the experimental results are shown in Fig. 10(c). Long flows will gradually lose throughput as the load increases, as is to be expected. However, due to the features of adaptive switching of transmission granularity and multi-path transmission, the throughput of long flows can be higher compared to previous load balancing schemes.

6.4. Performance with Incast traffic

Many distributed storage or computing activities, including Hadoop and MapReduce, are frequently present in the current datacenter work environment. Even if these computing services have a significant economic impact, the inescapable Incast communication mode causes several optimization difficulties for the real network transmission. A host makes data requests to numerous servers at once, and the server cluster responds at the same time. This is referred to as an incast, and it results in a significant rise in transmission traffic. Burst Incast will cause a lot of packets to get congested at the switch's outlet, which will lead to buffer overflow and high latency. How to handle the Incast communication mode is crucial for the datacenter with highly demanding low latency transmission.

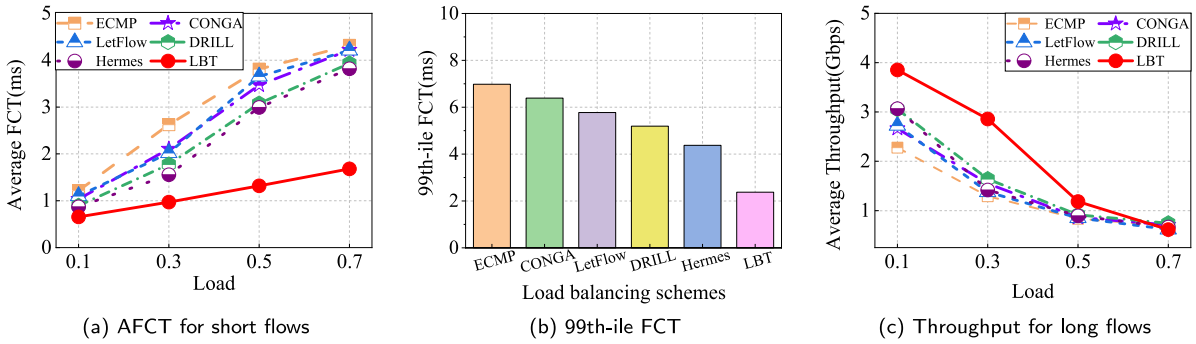
We use the parameter settings of NDP (Handley et al., 2017) in the simulated Incast experiment to verify the transmission performance of LBT in the Incast communication mode. One long-lived flow is sent by host 1, and one long-lived BE flow is transmitted by host 2. Start a short-lived 64-to −1 incast traffic pattern, transmitting 900KB each incast flow to host 3. Additional environments match those described in the Evaluation section. As seen in Fig. 11, when the incast occurs, the throughput of LBT is seen to be on a declining trend. Yet, because LBT itself has the transmission benefit of adjusting to burst flow, its value is significantly larger than that of DCTCP as the congestion control transmission schemes. According to LBT, each traffic flow selects the path with the smallest queue length. Since the first hop occurs in cast, this enables LBT to quickly change the load and reduce congestion in hot spots. By efficiently avoiding traffic congestion on a single link, all traffic is forwarded to the shortest queue, decreasing packet loss and increasing throughput.

7. Related work

With the continuous expansion of the datacenter network scale, the significant increase of network bandwidth and the continuous enhancement of traffic burst, how to improve the data center network transmission performance is a critical issue. In recent years, there are many load balancing schemes (Milani and Navimipour, 2016; Toosi et al., 2017) have emerged to improve the network transmission performance of the datacenter. These schemes based on heterogeneous traffic transmission can be divided into different cases according to the transmission granularity. We will discuss and summarize the existing schemes and their advantages and disadvantages.

The schemes based on flow granularity transmission. The most classic ECMP (Zhang et al., 2014) scheme uses static hashes to disperse data flows to equivalent multipath. Because ECMP cannot sense congestion, its performance is greatly reduced. According to connection capacity, WCMP (Zhou et al., 2014) adds weight to ECMP and hashes traffic to each path separately. However, both of them are prone to hash collision. Hedera (Al-Fares et al., 2010) uses the central controller to reroute long flows encountering congestion, which solves the problem. OmniFlow (Wen et al., 2016) combines load balancing and flow control to dynamically adjust routing paths to make full use of bandwidth. The scheme based on flow granularity can also divide different routes according to different needs of long and short flows. OFload (Trestian et al., 2017) uses OpenFlow switches to plan different routes for long flows and short flows. Hermes (Zhang et al., 2017) makes different rerouting decisions for flows with different size according to path status and flow status. However, the flow granularity based routing method has a low utilization rate of multi-path, and it is inflexible for long flow switching paths.

The schemes based on packet granularity transmission. RPS (Dixit et al., 2013) utilizes all route resources by randomly dispersing each data packet onto the path. DRB (Cao et al., 2013) selects the path for each data packet by polling to avoid selecting the same path for



(a) AFCT for short flows (b) 99th-ile FCT (c) Throughput for long flows

Fig. 10. Compare the performance in Data Mining workload.

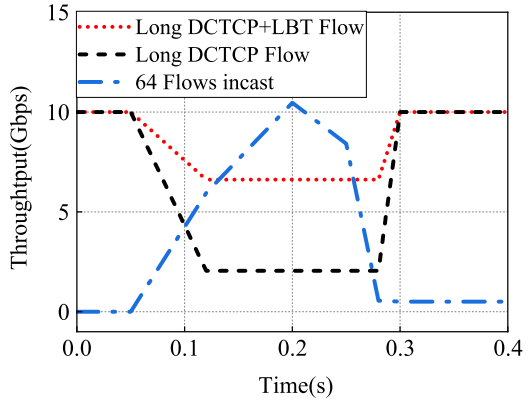


Fig. 11. Throughput comparison with incast traffic.

consecutive data packets. Drill (Ghorbani et al., 2017) selects the port with the lowest load as the packet forwarding port by comparing the queue length of two currently randomly selected ports and the port with the lowest load previously. Although the scheme based on packet granularity solves the problem of insufficient path utilization, it comes with the problem of out of order. Out of order in the network will cause the sending port to reduce the sending rate, thus reducing the throughput.

The schemes based on flowlet granularity transmission. CONGA (Alizadeh et al., 2014) selects the optimal next hop for traffic based on end-to-end path state real-time feedback of global information. Letflow (Vanini et al., 2017) distinguishes packet clusters according to a fixed time interval, and randomly selects a forwarding port for each cluster. However, FLARE stipulates that traffic rerouting occurs when the time interval between two clusters is greater than the maximum delay difference of the path. The scheme based on the flowlet granularity balances the packet granularity with the flow granularity, and makes a compromise between reordering and insufficient path utilization. However, the value of flow timeout is still controversial. When the flow timeout value is large, the traffic rerouting opportunities become less, and the path utilization is insufficient. When the flow timeout value is small, rerouting occurs frequently, and the disorder problem follows.

The schemes based on flowcell granularity transmission. Presto (He et al., 2015) divides the long and short flows into fixed units of 64 kb and iterates over multiple paths in a circular manner. Luopan (Wang et al., 2019) samples part of the paths regularly and forwards the fixed size packets to the path with the minimum queue length. Because such schemes always reroute with fixed cells, there will be problems of low flexibility and adaptability in the highly dynamic datacenter network environment.

The above four types of schemes with different granularity have their own advantages, but the above load balancing schemes have different research defects due to the same granularity transmission. In LBT, we are no longer limited to two types of traffic, but join BE flows. In addition, We set two thresholds to control the granularity of the transmission of long flows and BE flows and make the three types of flows route according to different routing methods. As a result, when long and short flows are on the same links, LBT ensures low latency for short flows and high throughput for long flows by resolving issues such packet disorder and long queue waiting for short flows.

8. Conclusion

In this paper, we propose LBT, a load balancing scheme for heterogeneous traffic. Its goal is to guarantee both the high throughput of long flows and the low latency of short flows. In order to switch the transmission granularity between long flows and BE flows adaptively, LBT specifically calculates two thresholds in accordance with various load intensities. Then, in accordance with the threshold, various traffic routing algorithms are created in order to give short flows more link resources and prevent long tail congestion. To increase throughput, long flows can flexibly choose pathways based on thresholds. In addition, BE flows can offer long and short flows more link resources based on their transmission characteristics. The NS-2 simulation results show that LBT reduces the AFCT of short flows by 55.9% ~ 65.4% with 0.5 load in comparison to the most complicated load balancing schemes under Data Mining.

This research still has some restrictions, though. Since no actual machine experiment has been performed, the experimental data in this study is based mostly on the NS-2 network simulation platform. The simulation experiment and the real experiment effect will be different, with inaccuracies. LBT also has the drawback of requiring a lot of switch storage resources. These are the issues that need further investigation. The use of memory and processing resources will then be minimized as much as possible. LBT will then be implemented on the programmable switch of the real datacenter network test platform, and more experiments will be run to determine the impact of LBT in the datacenter network.

CRediT authorship contribution statement

Jin Wang: Analysis, Manuscript preparation. **Shuying Rao:** Performed the NS-2 experiment, Data analyses, Wrote the manuscript . **Ying Liu:** Algorithm design. **Pradip Kumar Sharma:** Conception of the study. **Jinbin Hu:** Perform the analysis with constructive discussions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

References

Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A., 2010. Hedera: dynamic flow scheduling for data center networks. In: Nsdi. 10 (8), 89–92.

Alizadeh, M., Greenberg, A., et al., 2010. Data center TCP (DCTCP). In: Proc. ACM SIGCOMM. pp. 63–74.

Alizadeh, M., Yang, S., Sharif, M., Katti, S., McKeown, N., Prabhakar, B., Shenker, S., 2013. Pfabric: Minimal near-optimal datacenter transport. In: Proc. ACM SIGCOMM Conf. pp. 435–446.

Alizadeh, M., et al., 2014. CONGA: Distributed congestion-aware load balancing for datacenters. In: Proc. ACM Conf. SIGCOMM. pp. 503–514.

Anon, 2020. Multi-resource VNF deployment in a heterogeneous cloud. IEEE Trans. Comput. 71 (1), 81–91.

Benson, T., Akella, A., Maltz, D., 2010. Network traffific characteristics of data centers in the wild. In: Proc. ACM IMC. pp. 267–280.

Cao, J., Xia, R., Yang, P., Guo, C., Lu, G., Yuan, L., Zheng, Y., Wu, H., Xiong, Y., Maltz, D., 2013. Per-packet load-balanced low-latency routing for clos-based data center networks. In: Proc. ACM CoNEXT. pp. 49–60.

Chen, L., Chen, K., Bai, W., Alizadeh, M., 2016. Scheduling mix-flows in commodity datacenters with karuna. In: Proc. ACM SIGCOMM Conf. pp. 174–187.

Dixit, A., Prakash, P., Hu, Y.C., Kompella, R.R., 2013. On the impact of packet spraying in data center networks. In: Proc. IEEE INFOCOM. pp. 2130–2138.

Ghorbani, S., Yang, Z., Godfrey, P.B., Ganjali, Y., Firoozshahian, A., 2017. DRILL: Micro load balancing for low-latency data center networks. In: Proc. ACM SIGCOMM. pp. 225–238.

Handley, M., Raiciu, C., Agache, A., Voinescu, A., Moore, A.W., Antichi, G., Wójcik, M., 2017. Re-architecting datacenter networks and stacks for low latency and high performance. In: Proc. ACM SIGCOMM. pp. 29–42.

Hannabuss, S., 2012. Wiley encyclopedia of operations research and management science. Ref. Rev. 26 (2), 25–27.

He, K., Rozner, E., Agarwal, K., Felter, W., Carter, J., Akella, A., 2015. Presto: Edge-based load balancing for fast datacenter networks. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15). Association for Computing Machinery, New York, NY, USA. pp. 465–478.

He, X., Zheng, J., Dai, H., Zhang, C., Li, G., Dou, W., Rafique, W., Ni, Q., Chen, G., 2022. Continuous network update with consistency guaranteed in software-defined networks. IEEE/ACM Trans. Netw. 30 (3), 1424–1438.

Hu, J., Huang, J., Lv, W., Li, W., Li, Z., Jiang, W., Wang, J., He, T., 2021. Adjusting switching granularity of load balancing for heterogeneous datacenter traffic. IEEE/ACM Trans. Netw. 29 (5), 2367–2384.

Hu, J., Huang, J., Lv, W., Li, W., Wang, J., He, T., 2019a. TLB: Trafficaware load balancing with adaptive granularity in data center networks. In: Proc. ACM ICPP. pp. 1–10.

Hu, J., Huang, J., Lv, W., Zhou, Y., Wang, J., He, T., 2018. CAPS: Coding-based adaptive packet spraying to reduce flow completion time in data center. In: Proc. IEEE INFOCOM.

Hu, J., Huang, J., Lv, W., Zhou, Y., Wang, J., He, T., 2019b. CAPS: Coding-based adaptive packet spraying to reduce flow completion time in data center. In: IEEE/ACM Transactions on Networking. 27 (6), 2338–2353.

Jian, X., Wu, L., Yu, K., Aloqaily, M., Ben-Othman, J., 2021. Energy-efficient user association with load-balancing for cooperative IIoT network within B5G era. J. Netw. Comput. Appl. 189, 103110.

Kheirkhah, M., Wakeman, I., Parisis, G., 2016. MMPTCP: A multi-path transport protocol for data centers. In: Proc. IEEE INFOCOM. pp. 1–9.

Luo, L., Foerster, K.-T., Schmid, S., Yu, H., 2022. Optimizing multicast flows in high-bandwidth reconfigurable datacenter networks. J. Netw. Comput. Appl. 203, 103399.

Milani, A.S., Navimipour, N.J., 2016. Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. J. Netw. Comput. Appl. 71, 86–98.

Munir, A., Qazi, I.A., Uzmi, Z.A., Mushtaq, A., Ismail, S.N., Iqbal, M.S., Khan, B., 2013. Minimizing flow completion times in data centers. In: Proc. IEEE INFOCOM. pp. 2157–2165.

Toosi, A.N., Qu, C., Assunção, Buyya, R., 2017. Renewable-aware geographical load balancing of web applications for sustainable data centers. J. Netw. Comput. Appl. 83, 155–168.

Trestian, R., Katrinis, K., Muntean, G.-M., 2017. OFLoad: An OpenFlow-based dynamic load balancing strategy for datacenter networks. IEEE Trans. Netw. Serv. Manag. 14 (4), 792–803.

Vamanan, B., Hasan, J., Vijaykumar, T.N., 2012. Deadline-aware datacenter TCP (D2 TCP). In: Proc. of the ACM Special Interest Group on Data Communication (SIGCOMM), vol. 42 no.4. ACM, New York, pp. 115–126.

Vanini, E., Pan, R., Alizadeh, M., Taheri, P., Edsall, T., 2017. Let it flow: Resilient asymmetric load balancing with flowlet switching. In: Proc. USENIX NSDI. pp. 407–420.

Wang, J., Han, H., Li, H., He, S., Sharma, P.K., Chen, L., 2022. Multiple strategies differential privacy on sparse tensor factorization for network traffic analysis in 5G. IEEE Trans. Ind. Inform. 18 (3), 1939–1948.

Wang, J., Jin, C., Xiong, N., Tang, Q., Srivastava, G., 2021. Intelligent ubiquitous network accessibility for wireless-powered MEC in UAV-assisted B5G. IEEE Trans. Netw. Sci. Eng. 8 (4), 2801–2813.

Wang, P., Trimponias, G., Xu, H., Geng, Y., 2019. Luopan: Sampling based load balancing in data center networks. IEEE Trans. Parallel Distrib. Syst. 30 (1), 133–145.

Wen, K., Qian, Z., Zhang, S., Lu, S., 2016. OmniFlow: Coupling load balancing with flow control in datacenter networks. In: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). pp. 725–726.

Wilson, C., Ballani, H., Karagiannis, T., Rowtron, A., 2011. Better never than late: Meeting deadlines in datacenter networks. In: Proc. of the ACM Special Interest Group on Data Communication (SIGCOMM) vol. 41 no. 4. ACM, New York, pp. 50–61.

Xu, H., Li, B., 2014. RepFlow: Minimizing flow completion times with replicated flows in data centers. In: Proc. IEEE INFOCOM. pp. 1581–1589.

Zhang, J., Bai, W., Chen, K., 2019. Enabling ECN for datacenter networks with RTT variations. In: Proc. ACM CoNEXT. pp. 233–245.

Zhang, H., Guo, X., Yan, J., Liu, B., Shuai, Q., 2014. SDN-based ECMP algorithm for data center networks. In: 2014 IEEE Computers, Communications and IT Applications Conference. pp. 13–18.

Zhang, Y., Kumar, G., Dukkipati, N., Wu, X., Jha, P., Chowdhury, M., Vahdat, A., 2022. Aequitas: admission control for performance-critical rpcs in datacenters. In: Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM '22). Association for Computing Machinery, New York, NY, USA, pp. 1–18.

Zhang, H., Zhang, J., Bai, W., Chen, K., Chowdhury, M., 2017. Resilient datacenter load balancing in the wild. In: Proc. ACM SIGCOMM. pp. 253–266.

Zhou, L., Chou, C.-H., Bhuyan, L.N., Ramakrishnan, K.K., Wong, D., 2018. Joint server and network energy saving in data centers for latency-sensitive applications. In: Proc. IEEE IPDPS. pp. 700–709.

Zhou, J., Tewari, M., Zhu, M., Kabbani, A., Poutievski, L., Singh, A., Vahdat, A., 2014. WCMP: weighted cost multipathing for improved fairness in data centers. In: Proceedings of the Ninth European Conference on Computer Systems (EuroSys '14). Association for Computing Machinery, New York, NY, USA, pp. 1–14.

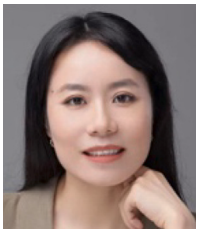
Jin Wang received the M.S. degree from Nanjing University of Posts and Telecommunications, China in 2005. He received Ph.D. degree from Kyung Hee University Korea in 2010. Now, he is a professor at Changsha University of Science and Technology. He has published more than 400 international journal and conference papers. His research interests mainly include wireless ad hoc and sensor network, network performance analysis and optimization etc. He is a senior member of the IEEE and a Fellow of IET.

Shuying Rao is currently pursuing the M.E. degree in the School of Computer and Communication Engineering at Changsha University of Science and Technology, China. Her research interests are in the area of datacenter networks and network security.

Ying Liu is currently pursuing the M.E. degree in the School of Computer and Communication Engineering at Changsha University of Science and Technology, China. His research interests include datacenter networks and evolutionary computation.

Dr. Pradip Kumar Sharma (M'18 SM'21) is an Assistant Professor of Cybersecurity in the Department of Computing Science at the University of Aberdeen, UK. He received his Ph.D. in CSE (August 2019) from the Seoul National University of Science and Technology, South Korea.

He has published many technical research papers in leading journals from IEEE, Elsevier, Springer, MDPI, etc. Some of his research findings are published in the most cited journals. He has been an expert reviewer for IEEE Transactions, Elsevier, Springer, and MDPI journals and magazines. He is listed in the world's Top 2% Scientists for citation impact during the calendar year 2019 by Stanford University. Also, he received a top 1% reviewer in computer science by Publons Peer Review Awards 2018 and 2019, Clarivate Analytics. He has also been invited to serve as the technical programme committee member and chair in several reputed international conferences such as IEEE DASC 2021, IEEE CNCC 2021, CSA 20202, IEEE ICC2019, IEEE MENACOMM'19, 3ICT 2019, etc. Currently, he is Associate Editor of Peer-to-Peer Networking and Applications (PPNA), Human-centric Computing and Information Sciences (HCIS), Electronics (MDPI), and Journal of Information Processing Systems (JIPS) journals. He has been serving as a Guest Editor for international journals of certain publishers such as IEEE, Elsevier, Springer, MDPI, JIPS, etc. His current research interests are focused on the areas of Cybersecurity, Blockchain, Edge computing, SDN, and IoT security.



Jinbin Hu received the B.E. and M.E. degrees from Beijing Jiao Tong University, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from Central South University, China, in 2020. She is currently a Post-Doc in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and working in the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. Her current research interests are in the area of datacenter networks, RDMA networking and learning-based network systems.