IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 8, 2021, accepted February 24, 2021, date of publication March 1, 2021, date of current version March 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062866

Coding-Based Distributed Congestion-Aware Packet Spraying to Avoid Reordering in Data Center Networks

JINBIN HU[1], **(Member, IEEE), CHANG RUAN**[1], **LEI WANG**[2], **OSAMA ALFARRAJ**[3], **AND AMR TOLBA**[3,4]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
[2]School of Civil Engineering, Changsha University of Science and Technology, Changsha 410114, China
[3]Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[4]Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shibin Al Kawm 32511, Egypt

Corresponding author: Lei Wang (leiwang@csust.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFC1511000, and in part by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia through the Research Group under Grant RG-1438-070.

ABSTRACT Modern Data Center Networks (DCNs) are commonly based on Clos topologies with a large number of equal-cost multiple paths to provide high bisection bandwidth. The existing Random Packet Spraying (RPS) scheme spreads each flow of packets to all available parallel paths in order to achieve good load balancing under symmetric topologies. However, under asymmetric topologies caused by traffic dynamics or link failures, RPS potentially suffers from serious out-of-order problem. Therefore, to avoid packet reordering, we propose a Coding-based Distributed Congestion-aware Packet Spraying mechanism called CDCPS. At the sender end, CDCPS encodes packets using forward error correction (FEC) technology and adaptively adjusts the coding redundancy according to the asymmetric degree of multiple equal-cost paths. To make full use of link bandwidth, CDCPS randomly spreads encoded packets to all available paths at the switches. The original packets can be recovered immediately once enough encoded packets from uncongested paths arrive at the receiver, even if some encoded packets are blocked on congested paths. The test results of NS2 simulation showed that CDCPS eliminates out-of-order packets completely and effectively reduces the average and 99^{th} flow completion time by up to 73% and 78% over the state-of-the-art load balancing scheme.

INDEX TERMS Asymmetry, data center network, load balancing, multiple path, packet spraying.

I. INTRODUCTION

To support the increasing traffic demands of large-scale distributed applications such as Web search, Data Mining and Machine Learning, modern Data Center Networks are typically based on Clos topologies, which provide high bisection bandwidth via a large number of equal-cost multiple paths between any pair of end-hosts [1], [2]. While path diversities widely exist in practice among the rich parallel paths for a variety of reasons, such as traffic dynamics, link failures and heterogeneity in network equipments [3], [4]. Therefore, some paths are congested with large queueing delay even some other paths are under-utilized, leading to topology asymmetric [5]–[7].

As a packet-level load balancing mechanism, Random Packet Spraying [8] is proposed to make full use of the multiple parallel paths in data centers. RPS randomly splits and spreads each flow into packets to one of the available paths to the destination. Under the symmetric topologies, RPS obtains high link utilization and achieves the best load balancing effect. In recent years, RPS has already been deployed on the commodity switches due to its simplicity [9].

However, due to lacking visibility into path congestion, RPS performs poorly in asymmetric topologies [3]. The key issue is the Transmission Control Protocol (TCP) out-of-order problem. Specifically, when packets that belong to a flow are assigned to different paths with different latencies,

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

the out-of-order event happens, in which the later-sent packets arrive at the receiver ahead of the earlier-sent ones. After receiving three duplicate acknowledgements (DupACKs) for the same packet, the TCP sender assumes the out-of-order packet is lost. Then TCP triggers congestion avoidance by cutting down the congestion window, resulting in spurious retransmission and even timeout.

In this paper, to adapt to asymmetry in the network topology, we propose a coding-based distributed congestion-aware packet spraying mechanism called CDCPS, which successfully integrates coding into packet spraying and completely avoids packet reordering. At the sender, CDCPS encodes packets of a flow using forward error correction (FEC) technology. Though some encoded packets experience queueing delay on the congested paths, once sufficient encoded packets from uncongested paths arrive at the receiver, the original source packets can be recovered immediately. To resilient to asymmetry, CDCPS is sensitive to path congestion and adaptively adjusts the coding redundancy according to the asymmetric degree of multiple paths. Specifically, CDCPS increases the coding redundancy under high degrees of topology asymmetry to avoid the influence of queued packets, and reduces the coding redundancy otherwise. We implement CDCPS between the TCP and Internet Protocol (IP) layers at the end hosts, while making no modifications on the TCP/IP protocol stack.

The main work of this paper contains three parts: (1) model and analysis of the impacts of out-of-order problem caused by RPS; (2) method of coding mechanism; and (3) performance evaluation through NS2 simulation tests. Considering the asymmetric network topology due to the dynamic traffic, CDCPS optimizes the coding redundancy based on the path congestion information such as Round Trip Time (RTT) to effectively avoid packet reordering and reduce flow completion time. The contributions of our proposed coding-based packet spraying scheme are mainly concentrated in three points.

- We exploit the impact of packet reordering in RPS load balancing mechanism and demonstrate through experiment and in theoretical way why the out-of-order probability is increased when the asymmetric degree of multiple paths increases.
- We propose a coding-based distributed congestion-aware packet spraying mechanism to handle asymmetric topology in data center networks. Our design encodes the packets of a flow and then randomly spreads the encoded packets to all parallel paths. By adaptively adjusting the coding redundancy according to the differences of path latency measured at the end-hosts, the design successfully avoids the impacts of reordering.
- We conduct large-scale NS2 simulations to evaluate our design under the realistic Web search and Data Mining workloads. The results demonstrated that, as with our design, we are able to effectively reduce the average and 99^{th} flow completion time.

The rest of the paper is organized as follows. Section II compares the related works. Section III describes motivation and analyzes the impact of packet reordering on the performance of RPS scheme. Section IV gives the design overview of Coding-based Distributed Congestion-aware Packet Spraying mechanism. Section V introduces the design details of Coding-based Distributed Congestion-aware Packet Spraying mechanism. Section VI shows the Performance evaluation and simulation results. Section VII presents the Conclusion and Future work.

II. RELATED WORK

Modern data center networks are organized as multi-rooted tree topologies to provide high bisection bandwidth. A rich body of load balancing mechanisms are proposed to fully utilize multiple equal-cost paths. We classify these load balancing approaches into four categories (i.e., flow-based, flowlet-based, flowcell-based and packet-based schemes) respectively.

A. FLOW-BASED SCHEMES

As the standardized flow-based scheme in data centers, Equal Cost Multi-Path (ECMP) [10] transfers each flow by using flow hash, thus suffers from hash collisions problem. To solve this problem, several enhanced flow-based load balancing schemes were proposed. Hedera [11] dynamically schedules long flows to uncongested paths by using a central controller to alleviate traffic hotspots. Like Hedera [11], MicroTE [12] also uses a central controller to assign flows by leveraging the partial predictability of traffic matrix. FlowBender [13] switches forwarding paths for flows once detecting the congestion or link failures to balance traffic. Hermes [4] timely and cautiously forwards short flows at flow level and reroutes long flows at packet level after gathering path congestion information. The above flow-based load balancing schemes have no out-of-order packets, however, they potentially suffered from long tail latency or low link utilization problems due to inability and no flexibility to change forwarding paths.

B. FLOWLET-BASED SCHEMES

Motivated by the drawbacks of flow-based schemes, many congestion-aware flowlet-based load balancing schemes are proposed. Once the interval time between two consecutive packets of a flow is larger than the preset threshold, a flowlet occurs and it can be rerouted. CONGA [1] estimates the congestion on each parallel paths from source leaf switch to destination leaf switch and then forwards each flowlet to the least congested path. LetFlow [3] randomly picks a path for each flowlet to naturally balance traffic, because the flowlet size changes automatically according to the path congestion degree. Flowtune [14] uses a centralized controller to allocate an optimal transmission rate for each flowlet. Clove [15] uses the virtual switches at the end-hosts to pick paths for flowlets in a weighted round-robin scheduling manner. HULA [16] assigns each flowlet to the least congested path based on the hop-by-hop congestion information at the programmable

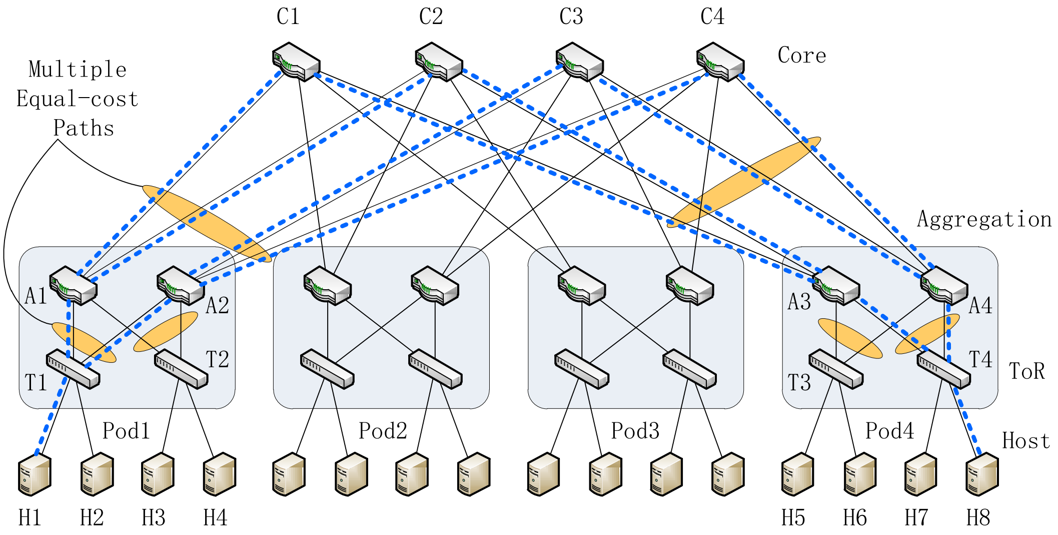


FIGURE 1. Fat-tree network topology.

switches. However, to completely avoid out-of-order packets in the above flowlet-based schemes, the flowlet timeout must larger than the maximum delay difference among the parallel paths, potentially leading to link underutilization.

C. FLOWCELL-BASED SCHEMES

To simultaneously improve link utilization and reduce out-of-order packets, Presto [17] performs load balancing at flowcell granularity with fixed 64KB. The flowcells are forwarded in a round-robin way. Luopan [18] reroutes flowcells to the least congested path of two randomly sampled paths. However, these flowcell-based schemes still suffered from packet reordering problem under the scenario with path diversities. Presto technique needs to reassemble the out-of-order flowcells at the receiver.

D. PACKET-BASED SCHEMES

To fully utilize the multiple paths, more fine-grained load balancing mechanisms are proposed. RPS [8] randomly spreads all packets to all available paths to achieve high link utilization. MMPTCP [19] makes switching decisions at packet level for short flows to reduce transmission time, and forwards long flows by using MPTCP [20] to achieve high throughput. By using a method similar to the power of two choices paradigm, DRILL [5] quickly and flexibly selects forwarding output port for each packet according to the local congestion information at switches. However, since these packet-level load balancing mechanisms are not aware of the path congestion information, they potentially suffer from packet reordering under the asymmetric topologies. To be resilient to asymmetric scenarios, AG [6] adaptively adjusts the path switching granularity based on the asymmetric degree of multiple paths. However, AG is not able to avoid out-of-order packets completely, when path diversity occurs.

In brief, it is hard for all above load balancing schemes to achieve no out-of-order packets and high link utilization simultaneously, especially under the asymmetric scenarios. Compared with those works, our solution CDCPS performs well at the packet granularity level in both symmetric and asymmetric topologies. CDCPS uses FEC coding technology to avoid packet reordering problem and uses RPS technology to obtain high link utilization by balancing traffic among multiple paths. Meanwhile, CDCPS adaptively adjusts the coding redundancy according to the asymmetric degree of parallel paths to handle asymmetry gracefully.

III. MOTIVATION

In this section, we firstly describe the drawback of RPS in the asymmetric topology and investigate the impact of packet reordering with RPS load balancing mechanism. Then we demonstrate theoretically why the reordering probability is increased when the asymmetric degree of multiple paths increases.

A. IMPACT OF PACKET REORDERING

Data center networks enable multiple equal-cost paths between a source-destination pair to simultaneously transfer packets. However, due to the link failures, heterogeneous switches or significant spatial and temporal variation of datacenter traffic [21]–[24], multiple parallel paths between host pairs are likely to exhibit different queueing buildup and path latency, resulting in asymmetric topology. Such asymmetry in the network topology potentially results in out-of-order packets in packet-level load balancing mechanisms.

Random packet spraying scheme distributes traffic equally over multiple parallel paths at packet granularity. As shown in Fig. 1, a flow from $H1$ to $H8$ that traverses the four paths $H1 \rightarrow T1 \rightarrow \{A1,A2\} \rightarrow \{C1, C2,C3,C4\} \rightarrow \{A3,A4\} \rightarrow T4 \rightarrow H8$ to reach the destination in a multi-rooted

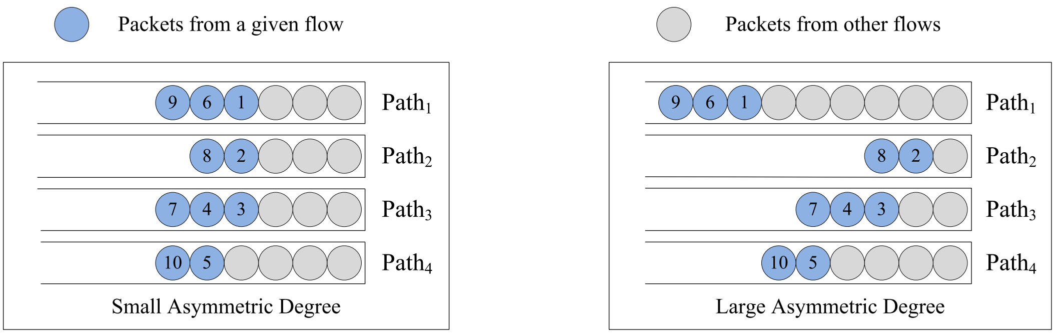
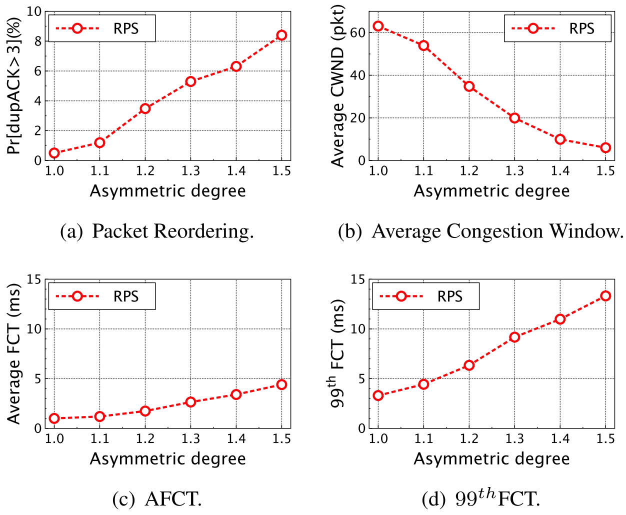


FIGURE 2. **Queueing buildup on equal-cost multiple paths.**

fat-tree topology. However, packet spraying leads to severe packet reordering in the case of path delay diversity. That means when the packets belong to a flow are assigned to different paths with different latencies, some packets arrive at the receiver out of order. Unfortunately, packet reordering problem is negatively interact with TCP congestion control. Since TCP is not able to distinguish out-of-order packets from lost packets, it will reduce congestion window size, resulting in throughput degradation.

Moreover, the performance of RPS is adversely affected by the degree of asymmetric topology. On the one hand, when the latency differential among multiple equal-cost paths is too large, the degree of topology asymmetry is high. Thus, the amount of induced out-of-order packets due to packet spraying is likely to be large. The out-of-order degree (OOD), which is defined as the maximal difference between the sequence number of an out-of-order packet and the expected packet sequence number, also becomes large, resulting in more buffered packets at the receiver. At the sender, TCP assumes a loss event has occurred after receiving three duplicate ACKs and performs retransmission at once. What's worse, more retransmission packets continuously injected into the network aggregate congestion, leading to a significant increase of the flow's tail latency. On the other hand, when the multiple parallel paths have similar latency, the amount of out-of-order packets becomes quite small under such low degree of topology asymmetry.

We use a simple example to illustrate the impact of packet reordering under different degrees of asymmetric topology. In Fig. 1, there are four equal-cost paths (i.e., *Path*$_1$ $\{H1, T1, A1, C1, A3, T4, H8\}$, *Path*$_2$ $\{H1, T1, A1, C2, A3, T4, H8\}$, *Path*$_3$ $\{H1, T1, A2, C3, A4, T4, H8\}$ and *Path*$_4$ $\{H1, T1, A2, C4, A4, T4, H8\}$) between the hosts $H1$ and $H8$. Fig. 2 shows the queueing buildup of ten packets from a given flow on the four equal-cost paths under different asymmetric degrees. The left figure in Fig. 2 shows the case of small asymmetric degree. Since the difference between the maximum and minimum queue lengths is one packet, only one packet with sequence number 8 arrives at the destination before the packet with sequence number 7, the out-of-order degree is 1. In the right figure in Fig. 2, with large asymmetric degree, the maximal difference in the queue length of the four equivalent paths is 5 packets. The later-sent packets with sequence numbers 2, 3, 4, 5, 7, 8, 10 reach the destination host $H8$ before the earlier-sent packet with sequence number 1. The amount of packets arrived at the destination host out of order is significantly increased compared with the case of low asymmetric degree. The out-of-order degree is increased to 9. In brief, RPS leads to severe packet reordering when the packets of a flow are routed over multiple parallel paths with large latency difference.



(a) Packet Reordering.

(b) Average Congestion Window.

(c) AFCT.

(d) 99thFCT.

FIGURE 3. **The Impact of packet reordering under asymmetric topology.**

We have conducted NS2 simulation test, in order to investigate the performance of RPS with varying asymmetric degree. The test topology is a typical fat-tree topology with 4 pods as shown in Fig. 1. There are 4 equal-cost paths between any pair of hosts in different pods. The bottleneck link bandwidth is 40Gbps and the round-trip propagation delay is 100μs. The switch buffer size is 256 packets. We generate 1000 flows between random pairs of hosts. All flows follow a Poisson process with flow size distribution of web search workload [9]. The test results show the average value of 100 runs.

In order to generate asymmetric topology, we increase the propagation delay of one path among four equivalent paths. The ratio of maximum RTT to minimum RTT among the four

TABLE 1. Variables in CDCPS.

Symbols	Definitions
S	Flow size in packets
n	Number of parallel paths
n_g	Number of uncongested paths
n_b	Number of congested paths
RTT_g	Path delay of uncongested paths
RTT_b	Path delay of congested paths
P_r	Reordering probability
P_g	Probability of a packet selecting uncongested path
P_b	Blocking probability of a packet selecting congested path
m_L	Number of ACKs with RTT larger than the average RTT
m	Total number of ACKs
x	Number of source packets in a coding unit
α	Number of redundant packets for a coding unit
P_d	Successful decoding probability

paths is defined as the asymmetric degree, which varies from 1 to 1.5. Fig. 1 (a) shows the ratio of more than 3 DupACKs events caused by out-of-order packets to all packets. With the increase of asymmetry degree, the ratio of reordering packets also increases. When the number of DupACKs reaches the retransmission threshold (default 3), TCP cuts the congestion window in half. Thus, the average congestion window decreases significantly as shown in Fig. 1 (b), degrading the performance of RPS. As shown in Fig. 1 (c) and Fig. 1 (d), the average and 99th flow completion time become larger with increasing degree of topology asymmetry. The reason is that RPS introduces more out-of-order packets under the scenario where the path delay difference becomes larger.

B. MODEL ANALYSIS

The degree of asymmetry affects the TCP reordering probability. In the following, we further analyze the reordering probability of existing RPS load balancing mechanism.

When the packets belong to a flow are assigned on multiple equal-cost paths, a packet is out of order only when at least one later-sent packet arrives at the destination host before the earlier-sent packets. Let S denote the size of a flow in packets. We assume that S packets are routed on n parallel paths, which includes n_g uncongested paths and n_b congested paths with the round trip latency RTT_g and RTT_b, respectively. In this situation, an out-of-order event occurs when one packet is transferred on a congested path and at least one later-sent packet is transferred on the uncongested path. Let P_g and P_b denote the probabilities of a packet selecting uncongested and congested paths, respectively. Table 1 shows all variables in CDCPS.

The reordering probability P_r of S packets in a flow is calculated as

$$P_r = \sum_{i=1}^{S-1} P_g^{i-1} \times P_b \times (1 - P_b^{S-i}). \tag{1}$$

Since every packet is randomly assigned to one of the available paths to the destination, the probability of a packet selecting uncongested path P_g is calculated as $\frac{n_g}{n}$. Thus, the probability of a packet selecting congested path P_b is obtain as

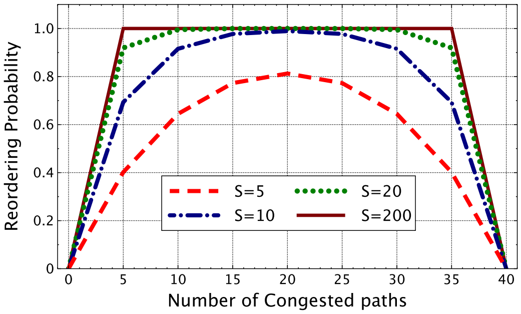


FIGURE 4. Reordering probability.

$$P_b = 1 - P_g = 1 - \frac{n_g}{n}. \tag{2}$$

Substitute P_g and P_b into Equation (1), we obtain the reordering probability P_r as

$$P_r = \sum_{i=1}^{S-1} (\frac{n_g}{n})^{i-1} \times (1 - \frac{n_g}{n}) \times (1 - (1 - \frac{n_g}{n})^{S-i}). \tag{3}$$

Next, we use the numerical analysis to show the change of reordering probability with the number of congested paths. The total number of parallel paths is set to 40. The number of congested paths varies from 0 to 40. Fig. 4 shows the reordering probability P_r with increasing number of congested paths n_b and the flow size S. Since the degree of topology asymmetry increases as the number of congested paths increases to half of the total paths, packet spraying causes more out-of-order packets. When half of the paths become congested, the reordering probability is the largest, resulting in the most serious packets reordering. In addition, with the increase of flow size, the number of out-of-order packets increases, leading to larger reordering probability. In brief, the existing RPS load balancing mechanism inevitably introduces packet reordering in the asymmetric scenarios. This conclusion motivates us to investigate a new approach to tolerant packet reordering under asymmetric topology in data center networks.

IV. DESIGN OVERVIEW

In this section, we presented the architecture of our proposed CDCPS mechanism. Our goal is to design a packet spraying load balancing mechanism adjusting coding redundancy based on the latency difference among multiple equal-cost paths in order to eliminate out-of-order packets completely and achieve high link utilization. When the delay difference of multiple paths is large, the amount of redundant encoded packets is increased. Therefore, more encoded packets are transferred on uncongested paths to eliminate the impact of packet reordering. On the contrary, when the delay difference becomes small, the amount of coding redundancy tends to decrease in order to adapt slight packet reordering. Fig. 5 shows the overview of our proposed CDCPS mechanism.

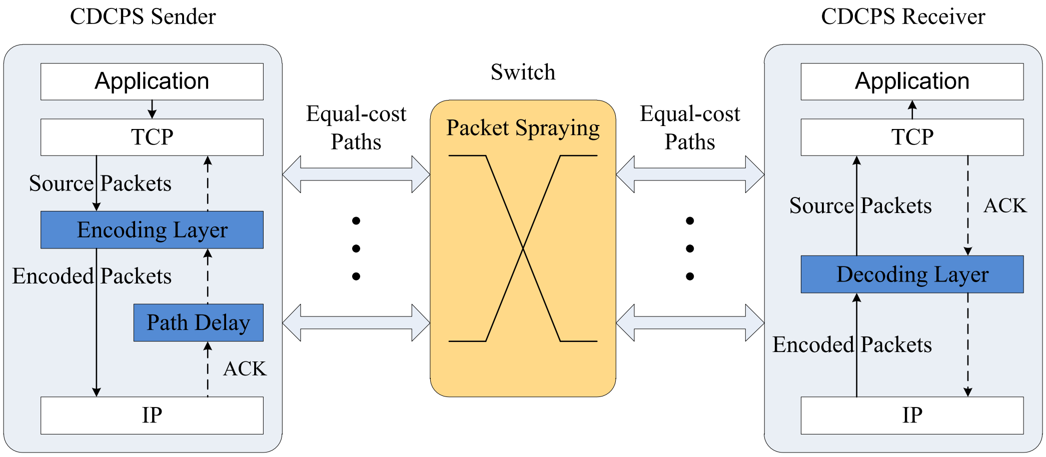


FIGURE 5. The overview of our proposed CDCPS mechanism.

(1) **At the sender:** The proposed distributed congestion-aware load balancing mechanism i.e. CDCPS estimates the number of congested paths based on the measurement of path delay (i.e., RTT). The sender generates source based encoded packets which are sent from the transport layer based on the real-time asymmetric degrees and delivers the encoded packets to the network layer.

(2) **At the switch:** The encoded packets are randomly spread to multiple equal-cost paths by using RPS technique, which has already been deployed in common commodity switches. Therefore, some encoded packets are blocked on congested paths with RTT larger than the average RTT, and some other encoded packets transferred on uncongested paths without blocking are able to arrive the destination host quickly.

(3) **At the receiver:** Even if some encoded packets are blocked on the congested paths, as long as enough encoded packets from uncongested paths arrive at the receiver, the original source packets can be recovered immediately, resulting in no out-of-order packets. Then the source packets are handed over to the upper layers.

V. DESIGN DETAILS

In this section, we firstly introduce our selected coding algorithm in proposed CDCPS load balancing mechanism. Then we discuss the coding redundancy optimization based on the latency differences among multiple parallel paths.

A. CODING ALGORITHM

Coding is a very powerful scheme to solve spurious retransmission and timeout problems caused by packet reordering as it can tolerate out-of-order packets for TCP. The key point of coding for TCP is to perform Forward Error Correction (FEC). With FEC technology incorporating into TCP [25], [26], the sender encodes source packets in a redundant way and transmits encoded packets via multiple equal-cost paths. Then the receiver decodes the original packets if enough encoded packets are received without waiting for the blocked packets or retransmission. Since FEC only cares about how many rather than which encoded packets have been received, FEC effectively eliminates the negative impact of packet reordering.

In our proposed CDCPS mechanism, we use the fixed-rate coding scheme [9] to generate encoded packets due to the following reasons. Firstly, the number of congested paths can be estimated in advance at the sender based on the measurement of path delay. The blocking probability of a packet selecting congested path can be estimated as the ratio of the number of paths with large RTT to the total number of paths. However, it is hard for CDCPS to directly obtain the accurate RTT for every path. Here, we take the advantages of TCP congestion control mechanism. Specifically, when the sender receives ACK packets, based on the corresponding RTT for each ACK packet, the blocking probability of a packet selecting congested path is equivalently calculated as the ratio of the number of ACKs with RTT larger than the average RTT to the total number of received ACKs. Secondly, since the receiver does not need to send feedback information to the senders to stop encoding [27], the fixed-rate coding method avoids unnecessary redundant packets and increases the robustness of data transmission.

The encoded packets are the random linear combinations of the source packets. For example, as illustrated in Fig. 6, 4 source packets ($P1$,$P2$,$P3$,$P4$) are encoded into 6 encoded packets by using random linear coding at the sender. Then the encoded packets are randomly spread to multiple equal-cost paths by using RPS technology at the switch. In this case, two encoded packets are transferred and blocked on congested paths. However, once arbitrary four independent encoded packets arrive at the receiver, the four original source packets can be decoded successfully without waiting for the blocked packets. Note that if additional encoded packets are subsequently received, they are dropped directly at the receiver.

B. CODING REDUNDANCY OPTIMIZATION

Under the different asymmetric degrees of network topology, the reordering probability and the blocking probability

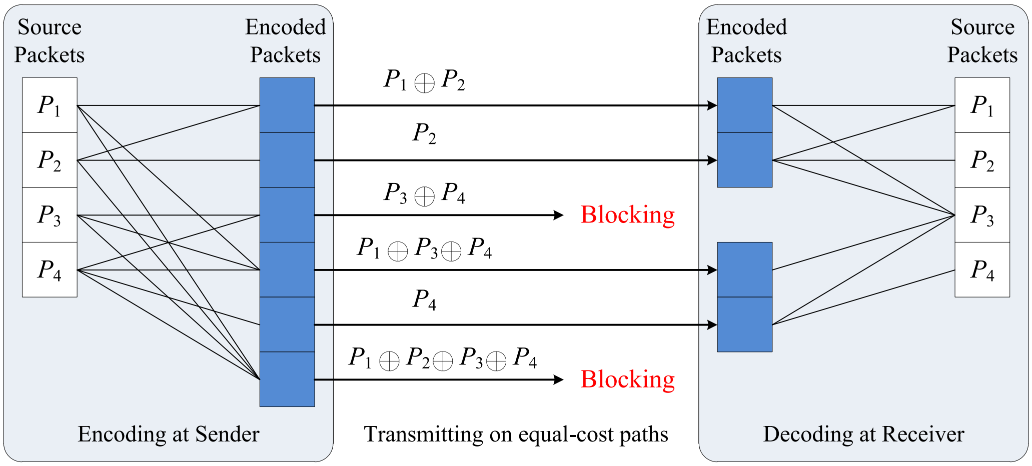


FIGURE 6. Example of coding.

analyzed in Section III-B are changed. Therefore, the coding redundancy affects traffic overhead and decoding delay. Specifically, when the difference of path latency is large, the time difference between consecutive packets belong to a flow arriving at the same destination is large. In this case, if too less redundant packets are transferred, the decoding speed is limited because there are not enough encoded packets reached at the receiver for decoding operation. It is necessary to increase the amount of redundant encoded packets to speed up the decoding operation. On the contrary, when the difference of path latency is small, too much redundant packets introduce unnecessary traffic overhead and limit the transmission rate of the sender by itself. In brief, to achieve good tradeoff between the decoding delay and traffic overhead, the coding redundancy should be dynamically adjusted according to the real-time blocking probability. In the following, we analyze the coding redundancy optimization.

As illustrated in Table 1 in Section III-B, m and m_L denote the total number of received ACKs and the number of ACKs with RTT larger than the average RTT, respectively. Then the blocking probability of a packet selecting a congested path is calculated as $P_b = \frac{m_L}{m}$. We assume x source packets in a coding unit are encoded into $x+\alpha$ encoded packets. To ensure that at least x encoded packets are transferred on uncongested paths without blocking, the following Equation (4) should be satisfied

$$(1 - P_b) \times (x + \alpha) \geq k. \tag{4}$$

To guarantee the lowest traffic overhead, substitute P_b into Equation (4), we obtain the number of redundant packets α for each x original packets as

$$\alpha = \lceil \frac{x}{1 - P_b} - x \rceil. \tag{5}$$

Fig.7 shows the coding redundancy α with varying blocking probability P_b. As the number of congested paths increases, the blocking probability is increased, leading to more redundancy packets for successfully decoding. However, since all encoded packets are randomly sprayed to multiple equal-cost paths, unavoidably blocking or even dropping some encoded packets, the decoding probability is not guaranteed 100%. We calculate the successful decoding probability P_s in the following. We assume that i encoded packets are transferred and blocked on the congested paths in a coding unit with $x + \alpha$ encoded packets. The probability of arbitrary blocked i encoded packets out of $x + \alpha$ encoded packets is calculated as

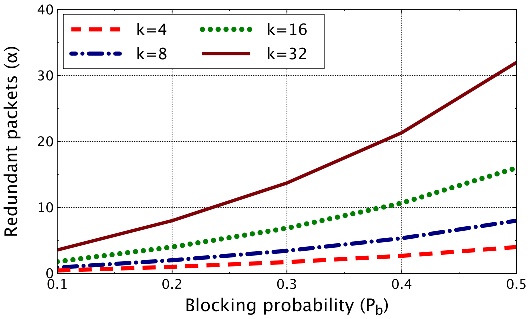


FIGURE 7. Number of redundant packets α with varying P_b and x.

$$P_b(i) = C^i_{x+\alpha} \times {P_b}^i \times (1 - P_b)^{x+\alpha-i}. \tag{6}$$

Once the number of encoded packets arrived at the receiver is no less than x, that means the number of blocked packets is no larger than the coding redundancy α, the source packets can be successfully decoded. We obtain the successful decoding probability P_d as

$$P_d = \sum_{i=0}^{\alpha} P_b(i) = \sum_{i=0}^{\alpha} C^i_{x+\alpha} \times {P_b}^i \times (1 - P_b)^{x+\alpha-i}. \tag{7}$$

Fig.8 shows the successful decoding probability P_d with varying redundancy. In this numeric analysis, the total number of paths and the coding unit x are set to 40 and 8, respectively. The blocking probability is calculated as the

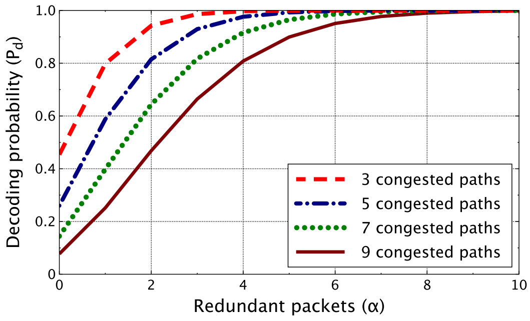


FIGURE 8. Successful decoding probability P_d with varying α and m_L.

ratio of the number of congested paths to the total number of paths. Less redundant packets make the successful decoding probability becomes lower. The reason is that some encoded packets are blocked on the congested paths, the receiver has to wait for enough encoded packets for decoding operation. In addition, with the increasing number of congested paths, more redundant packets are needed to make the successful decoding probability reaches 1.

VI. PERFORMANCE EVALUATION

In this section, we describe the conduction details of large-scale NS2 simulations of CDCPS technique, in order to evaluate the performance of CDCPS under two realistic datacenter workloads. We compared CDCPS with the state-of-the-art packet-level load balancing mechanisms in data centers. The performance metrics include the packet reordering, the average and 99^{th} tail flow completion time.

A. TEST TOPOLOGY

We build a typical leaf-spine topology with 8 leaf and 8 core switches. There are 8 equal-cost multiple paths between any pair of end-hosts. The whole network has 256 end-hosts connected by 40Gbps links. The round-trip propagation delay is 100μs. To generate asymmetric topology, we set the round-trip propagation delay of one randomly selected path to 500μs. The buffer size at switches is set to 256 packets.

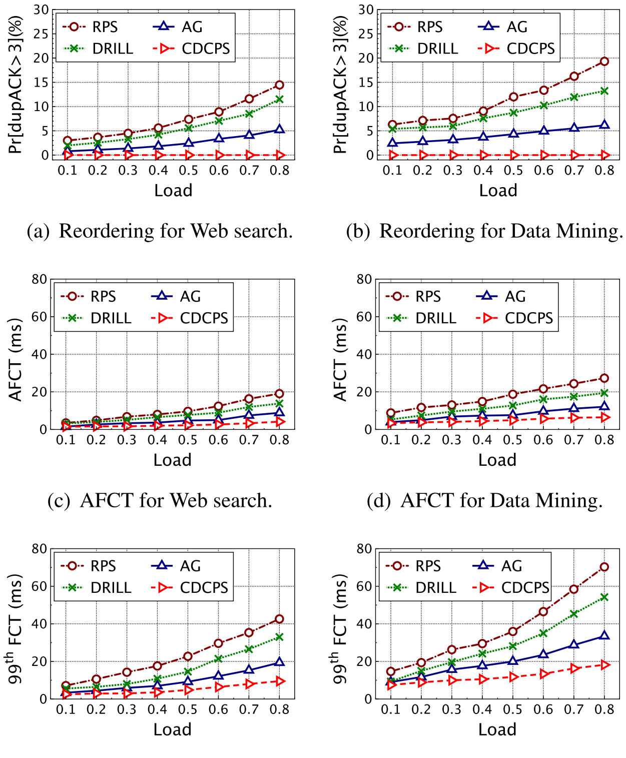
B. SCHEMES COMPARED

Besides RPS, we compared CDCPS with the following schemes.

- **DRILL [5]:** DRILL quickly and flexibly selects forwarding paths for each packet according to the local queueing information at switch. DRILL uses a method similar to the power of two choices algorithm [28] to make forwarding decisions. It compares the queue lengths of two random output ports and the last selected forwarding port, then chooses the one with the minimum queue length for the current forwarding port.
- **AG [6]:** AG dynamically adjusts the path switching granularity based on the asymmetric degree of network topology and then randomly spreads each packet train to one of the multiple equal-cost paths.

TABLE 2. Flow size distribution of realistic workload.

Workload	0-100KB	100KB-1MB	>1MB	Avg. flow size
Web search	62%	18%	20%	1.6MB
Data Mining	83%	8%	9%	7.41MB



(a) Reordering for Web search.

(b) Reordering for Data Mining.

(c) AFCT for Web search.

(d) AFCT for Data Mining.

(e) 99^{th} FCT for Web search.

(f) 99^{th} FCT for Data Mining.

FIGURE 9. Web search and Data Mining workload.

C. REALISTIC WORKLOADS

We took two widely-used realistic datacenter workloads: Web search [29] and Data Mining [30]. As shown in Table 2, the flow size distributions of both web search and data mining are heavy-tailed. Particularly, in the Web search workload, about 30% flows is seen larger than 1MB provided about 95% data bytes. In the Data Mining workload, about 3.6% flows is observed larger than 35MB, which provided about 95% data bytes. All flows are generated between random pair of hosts under different leaf switches according to Poisson processes. The overall workload varies from 0.1 to 0.8, in order to evaluate thoroughly the performance of CDCPS. We focused on the ratio of out-of-order packets and the flow completion time of all flows.

Fig.9 (c) and Fig.9 (d) show that CDCPS outperformed the other three schemes in the average flow completion time in both Web search and Data Mining workloads since CDCPS completely avoids out-of-order packets.

Fig.9 shows the simulation results of web search and data mining workloads. Fig.9 (a) and Fig.9 (b) show the ratio of more than 3 DupACKs events caused by packet reordering, Fig.9 (c) and Fig.9 (d) show the average flow completion time, while Fig.9 (e) and Fig.9 (f) give the 99^{th} FCT, presenting the tail flow completion time.

In Fig.9 (a) and Fig.9 (b), the ratio of out-of-order packets in CDCPS is always zero, because CDCPS utilizes the encoded packets from uncongested paths to recover the original source packets and eliminate the negative effect of packet reordering. The ratio of more than 3 DupACKs events caused by packet reordering increases with higher traffic load in RPS, DRILL and AG. The reason is that RPS and DRILL spread packets to all available paths with latency diversity, thus, some packets are blocked on the congested paths, leading to serious packet reordering especially under the higher traffic load.

Fig.9 (c) and Fig.9 (d) show that CDCPS outperforms the other three schemes in the average flow completion time in both Web search and Data Mining workloads since CDCPS completely avoids out-of-order packets. RPS performs poorly compared with other schemes, because RPS simply spreads all packets to all parallel paths without being aware of congestion and experiences the most serious packet reordering under the asymmetric topology. Under 0.6 traffic load, CDCPS reduces the AFCT of all flows in Web search and Data Mining by 73%, 66%, 41% and 69%, 60%, 38% over RPS, DRILL and AG, respectively.

Fig.9 (e) and Fig.9 (f) show that CDCPS achieves the lowest tail flow completion time. The reason is that CDCPS effectively avoids the out-of-order packets by using encoded packets. Moreover, compared with RPS, DRILL and AG, CDCPS can sense the latency differences of multiple paths. CDCPS adaptively adjusts the coding redundancy to be resilient to the asymmetric topology. Though DRILL and AG alleviate the negative impact of packet reordering, they are not adaptable to the high asymmetric degrees. For example, under the Web search workload, CDCPS reduces the 99^{th} FCT by 67% -78%, 62%-71%, 42%-51% for load from 0.3 to 0.8 over RPS, DRILL and AG, respectively.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new load balancing mechanism called CDCPS. The main motivation for introducing CDCPS is to completely avoid the negative impact of packet reordering under the asymmetric topologies in data center networks. CDCPS is a coding-based distributed congestion-aware packet spraying scheme. CDCPS encodes the packets of a flow at the sender, randomly spreads the encoded packets to all available multiple paths at switches, and then recovers the original source packets at the receiver once receiving enough encoded packets. CDCPS adaptively adjusts the coding redundancy according to path congestion information to resilient to topology asymmetry. The NS2 simulations, with realistic workloads representative of datacenter applications, show that CDCPS substantially reduces the average and tail latency compared to RPS, DRILL and AG. In particular, CDCPS respectively reduces the average and 99^{th} flow completion time by up to 73% and 78% compared to RPS.

In the future work, there are several problems remain to be well addressed. Firstly, the impact of coding redundancy on congestion control should be considered as the traffic overhead introduced by coding may aggregate the network congestion. Secondly, it is important to optimize the size of coding unit to simultaneously guarantee the high decoding efficiency and the low blocking probability of packets. Thirdly, comprehensive analysis and modeling for the overhead of encoding and decoding computing are needed to indicate the benefit of coding on latency. This paper has only discussed the scenario of path diversity where coding can eliminate packet reordering. More realistic application scenarios on the use of coding can be further explored.

REFERENCES

[1] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese, "CONGA: Distributed congestion-aware load balancing for datacenters," in *Proc. ACM SIGCOMM*, 2014, pp. 503–514.

[2] J. Hu, J. Huang, J. Lv, W. Li, J. Wang, and T. He, "TLB: Traffic-aware load balancing with adaptive granularity in data center networks," in *Proc. ACM ICPP*, 2019, pp. 1–10.

[3] E. Vanini, R. Pan, M. Alizadeh, P. Taheri, and T. Edsall, "Let it flow: Resilient asymmetric load balancing with flowlet switching," in *Proc. USENIX NSDI*, 2017, pp. 407–420.

[4] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury, "Resilient datacenter load balancing in the wild," in *Proc. ACM SIGCOMM*, 2017, pp. 253–266.

[5] S. Ghorbani, Z. Yang, P. B. Godfrey, Y. Ganjali, and A. Firoozshahian, "DRILL: Micro load balancing for low-latency data center networks," in *Proc. ACM SIGCOMM*, Aug. 2017, pp. 225–238.

[6] J. Liu, J. Huang, W. Li, and J. Wang, "AG: Adaptive switching granularity for load balancing with asymmetric topology in data center network," in *Proc. IEEE ICNP*, Oct. 2019, pp. 1–11.

[7] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. ACM IMC*, 2010, pp. 267–280.

[8] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella, "On the impact of packet spraying in data center networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2130–2138.

[9] J. Hu, J. Huang, W. Lv, Y. Zhou, J. Wang, and T. He, "CAPS: Coding-based adaptive packet spraying to reduce flow completion time in data center," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2338–2353, Dec. 2019.

[10] C. Hopps, *Analysis of an Equal-Cost Multi-Path Algorithm*, document RFC 2992, Internet Engineering Task Force, 2000.

[11] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. USENIX NSDI*, 2010, pp. 19–34.

[12] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine grained traffic engineering for data centers," in *Proc. ACM CoNEXT*, 2011, pp. 1–12.

[13] A. Kabbani, B. Vamanan, J. Hasan, and F. Duchene, "FlowBender: Flow-level adaptive routing for improved latency and throughput in datacenter networks," in *Proc. ACM CoNEXT*, Dec. 2014, pp. 149–160.

[14] J. Perry, H. Balakrishnan, and D. Shah, "Flowtune: Flowlet control for datacenter networks," in *Proc. USENIX NSDI*, 2017, pp. 421–435.

[15] N. Katta, A. Ghag, M. Hira, I. Keslassy, A. Bergman, C. Kim, and J. Rexford, "Clove: Congestion-aware load balancing at the virtual edge," in *Proc. ACM CoNEXT*, Nov. 2017, pp. 323–335.

[16] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "HULA: Scalable load balancing using programmable data planes," in *Proc. ACM SOSR*, Mar. 2016, p. 10.

[17] K. He, E. Rozner, K. Agarwal, W. Felter, J. Carter, and A. Akella, "Presto: Edge-based load balancing for fast datacenter networks," in *Proc. ACM SIGCOMM*, 2015, pp. 465–478.

[18] P. Wang, G. Trimponias, H. Xu, and Y. Geng, "Luopan: Sampling-based load balancing in data center networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 1, pp. 133–145, Jan. 2019.

[19] M. Kheirkhah, I. Wakeman, and G. Parisis, "MMPTCP: A multipath transport protocol for data centers," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[20] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," in *Proc. ACM SIGCOMM*, 2011, pp. 266–277.

[21] J. Huang, Y. Huang, J. Wang, and T. He, "Adjusting packet size to mitigate TCP Incast in data center networks with COTS switches," *IEEE Trans. Cloud Comput.*, vol. 8, no. 3, pp. 749–763, Jul./Sep. 2020.

[22] T. Zhang, J. Huang, K. Chen, J. Wang, J. Chen, Y. Pan, and G. Min, "Rethinking fast and friendly transport in data center networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2364–2377, Oct. 2020.

[23] S. Liu, J. Huang, Y. Zhou, J. Wang, and T. He, "Task-aware TCP in data center networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 389–404, Feb. 2019.

[24] J. Huang, S. Li, R. Han, and J. Wang, "Receiver-driven fair congestion control for TCP outcast in data center networks," *J. Netw. Comput. Appl.*, vol. 131, pp. 75–88, Apr. 2019.

[25] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 465–478, Apr. 2015.

[26] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, "Network coding meets TCP," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 280–288.

[27] O. C. Kwon, Y. Go, Y. Park, and H. Song, "MPMTP: Multipath multimedia transport protocol using systematic raptor codes over wireless networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1903–1916, Sep. 2015.

[28] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

[29] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *Proc. ACM SIGCOMM*, 2010, pp. 63–74.

[30] 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," in *Proc. ACM SIGCOMM*, 2009, pp. 51–62.

JINBIN HU (Member, IEEE) received the B.S. and M.S. degrees from Beijing Jiaotong University, China, in 2008 and 2011, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Central South University, China. She also works as a Teacher with the School of Computer and Communication Engineering, Changsha University of Science and Technology. Her current research interest includes data center networks.

CHANG RUAN was born in Hubei, China, in September 1986. He received the bachelor's, master's, and Ph.D. degrees from Central South University, China, in 2007, 2011, and 2017, respectively. He currently works as a Teacher with the School of Computer and Communication Engineering, Changsha University of Science and Technology. He also holds a postdoctoral position with the School of Information Science and Engineering, Central South University. His current research interests include data center networks and control theory.

LEI WANG received the Ph.D. degree from the School of Civil Engineering, Changsha University of Science and Technology, China, in 2008. He is currently a Professor with the School of Civil Engineering, Changsha University of Science and Technology. His research interests include the intelligent construction and maintenance of structures, the structural durability, and the structural reliability.

OSAMA ALFARRAJ received the master's and Ph.D. degrees in information and communication technology from Griffith University, in 2008 and 2013, respectively. He is currently an Associate Professor of computer sciences with King Saudi University, Riyadh, Saudi Arabia. His current research interests include eSystems (eGov, eHealth, and ecommerce), cloud computing, and big data. For two years, he has served as a Consultant and a member for the Saudi National Team for Measuring E-Government, Saudi Arabia.

AMR TOLBA received the M.Sc. and Ph.D. degrees from the Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently an Associate Professor with the Faculty of Science, Menoufia University. He is on leave from Menoufia University to the Computer Science Department, Community College, King Saud University (KSU), Saudi Arabia. He has authored/coauthored more than 75 scientific articles in top ranked (ISI) international journals and conference proceedings. His main research interests include socially aware networks, vehicular ad-hoc networks, the Internet of Things, intelligent systems, big data, recommender systems, and cloud computing. He serves as a technical program committee (TPC) member for several conferences.