# SoftFG: A Dynamic Load Balancer for Soft Reconfiguration of Wireless Data Centers

Amer AlGhadhban[§¶], Abdulkadir Celik[§], Basem Shihada[§], and Mohamed-Slim Alouini[§]
[§]*Computer, Electrical, Mathematical Sciences & Engineering (CEMSE) Division*
*King Abdullah University of Science and Technology (KAUST)*
Thuwal, Makkah Province, 23955, KSA.
[¶]*College of Engineering, Electrical Engineering Department*
*University of Ha'il*
Hail, 55476, KSA.
{amer.alghadhban, abdulkadir.celik, basem.shihada, slim.alouini}@kaust.edu.sa

***Abstract*—In this paper, we investigate the soft-reconfiguration of optical wireless data centers (WDCs). In the considered physical topology, edge top-of-rack (ToR) switches in the leaf layer are inter-connected with core switches in the spine layer via wavelength division multiplexing (WDM) based free-space optical (FSO) links. We propose an agile load balancing (LB) solution, namely *SoftFG*, to cope with the dynamically changing link load variations and the low-utilization time intervals within the wireless data centers (DCs). *SoftFG* executes flow grooming (FG) and soft reconfigurations on the virtual topology depending upon the fine-grain network statistics. Unlike the long-term LBs, SoftFG offloads large flows of congested paths onto underutilized links without making any hardware reconfiguration on path capacity and routes. Flows can be offloaded to other wavelengths within the same FSO link (i.e., intra-link), to other FSO links (i.e., inter-link), or within/across topologies (i.e., intra/inter topology). To do so, SoftFG ensures clear visibility on network paths, early congestion detection, and fast-accurate reaction to reroute offloaded flows onto underutilized wavelengths or links. Therefore, SoftFG is designed as a kernel module installed on the virtual switches/hypervisor. The module collects flow statistics based on a source-destination collaborative scheme and records them in flow and path information tables. SoftFG accordingly makes quick decisions on offloading and reroutes flows with high accuracy. Emulation results show that SoftFG delivers about 12× and 17× faster flow completion time (FCT) than LetFlow and CONGA LBs, respectively.**

***Index Terms*—Load balancing; Adaptive probing, Flow Grooming, Flow Detection.**

## I. Introduction

Data center networks (DCNs) encounters high stakes of bandwidth demands due to the proliferation of bandwidth-hungry technology trends such as big data, internet of things, artificial intelligence, and the fifth-generation (5G) networks [1]. Recent studies have found that the majority of traffic is exchanged between a few racks while remaining racks exchange less traffic or no traffic at all, which yields asymmetric utilization of network resources. Moreover, they have shown that existing wired DCN topologies, where the link capacity is fixed and uniform at each tier, cannot support the optimal capacity allocation mechanisms and not flexible for adapting to workload dynamics.

As a remedy, the state-of-the-art wireless technologies such as mmWave and free-space optical (FSO) communications have been considered to improve the performance of DCNs [2]. Replacing cables with wireless links enables reconfigurable DCN topologies, which can cope with the aforementioned dynamic traffic patterns and provide low cabling complexity and maintenance overhead. However, wireless DCNs are mostly studied in the domain of physical topology design that generally focuses on establishing line-of-sight (LoS) connectivity between the racks. LoS connectivity can be obtained by using ceiling mirrors [3], employing a disco-ball that is made of digital micromirror devices [4] or placing racks in cells where racks can see each other [5].

Alternatively, the authors of [6]–[8] replaced the ceiling mirror with hybrid-cross-connect (HXC) switches. Based on the so-called three-step flow grooming (3SFG) approach, mice flow (MF) traffic is groomed together and forwarded through predetermined rack-to-rack lightpaths, whereas elephant flows (EFs) are transmitted via express lightpaths without grooming. The 3SFG approach determines the path capacity and routes based on the first order workload statistics of the traffic among the racks. Hence, 3SFG can also be regarded as a long-term load balancer (LB). However, the 3SFG assumes that flow classes are known, which is indeed not readily available in practice. Moreover, the 3SFG is not capable of handling bursty traffic and short term variations in the traffic conditions. Although it is possible to enhance 3SFG by modifying the virtual topology more frequently based on the short term statistics, hard reconfigurations of topology as well as link capacity entail serious network intervention and flow preemptive scheduling.

On top of the 3SFG approach, we propose an agile LB solution to cope with the dynamically changing link load variations and the low-utilization time intervals. It is called *SoftFG* as it executes soft reconfigurations on the virtual topology depending upon the fine-grain network statistics. Unlike the long-term LB nature of the 3SFG, the SoftFG offloads large flows of congested paths onto underutilized links without making any hardware reconfiguration on path capacity and routes. Flows can be offloaded to other wavelengths within

the same FSO link (i.e., intra-link), to other FSO links (i.e., inter-link), or within/across topologies (i.e., intra/inter topology). To do so, SoftFG requires clear visibility on network paths, early congestion detection, and fast-accurate reaction to reroute offloaded flows onto underutilized wavelengths or links. Therefore, SoftFG is designed as a kernel module installed on the virtual switches/hypervisor. The module collects flow statistics based on a source-destination collaborative scheme and records them in flow and path information tables. SoftFG accordingly makes quick decisions on offloading and reroutes with high accuracy. Emulation results show that SoftFG delivers about $12\times$ and $17\times$ faster flow completion time (FCT) than LetFlow and CONGA LBs, respectively.

The rest of this paper is organized as follows: Section II gives an overview of the 3SFG based long-term LB. Then, Section III presents SoftFG along with implementation details. Section IV provides emulation results. Finally, Section V concludes the paper with a few remarks.

## II. 3SFG as a Long-Term Load Balancing Scheme

The 3SFG approach considers two virtual topologies on a spine-leaf layer physical topology where edge switches (ESs) (i.e., racks) in the leaf layer are inter-connected with core switches in the spine layer via wavelength division multiplexing (WDM) based FSO links [6], [8]. Since forwarding MFs and EFs on the same paths severely degrades the DCN performance, each virtual topology is dedicated to a particular flow class and isolated from the other.

Assuming the availability apriori flow class information, the 3SFG executes server-to-server (S2S), server-to-rack, and rack-to-rack (R2R) MF grooming to obtain R2R-MFs, which are then directed to the relevant optical transmitter based on the predetermined routing paths. The capacity allocation to R2R-MFs is determined based on the long-term statistics of the entire traffic across the servers within the source and destination racks. On the other hand, route determination is jointly implemented with the capacity allocation such that the number of R2R- MFs and resulting capacity load is balanced by distributing the R2R-MF lightpaths across all available FSO links. To do so, predetermined R2R-MF lightpaths are iteratively assigned on the routes with the maximum number of available wavelengths. That is, R2R-MF lightpath routes ensure that the number of available wavelengths and light intensity on FSO links are evenly distributed across the entire topology. Although 3SFG routes EFs overexpress (S2S) paths without any grooming procedure, we consider a modified version by allowing EF grooming to obtain R2R-EFs. Following R2R-MF lightpath provisioning, the residual light intensity and available wavelengths of FSO links are then exploited by the R2R-EFs based on the previous joint intensity allocation and route determination method.

Along with this routing and resource allocation approach, the 3SFG behaves as a long-term LB and efficiently utilizes the available bandwidth while provisioning the QoS demands of each flow class at the same time. Since the 3SFG based load balancing mainly depends on the first order traffic statistics, it is not capable of reaping the full benefits of under-utilized links and time intervals, which motivated us to propose an agile and accurate short-term LB scheme, SoftFG.

## III. SoftFG: An Agile Short-Term Load Balancing

An efficient short-term LB mainly requires clear visibility on network paths, early congestion detection, and fast-accurate rerouting decisions. However, existing LB schemes are tailored to fixed-wired DCNs, hence, not suitable to tackle the dynamically changing wireless DCN topologies. To the best of authors' knowledge, SoftFG is the first to address challenges of LB in wireless DCNs with several virtual topologies dedicated to different flow classes, where several challenges arises:

1) The lightpath capacity in virtual-topology is configured to match the workload distribution that yields asymmetric lightpath capacities on the same FSO link.
2) Due to the segmentation of the dedicated topologies, the flows require careful rerouting across different topologies.
3) The rack cannot maintain enough visibility on every rack-to-rack paths, which hinders an efficient rerouting mechanism.

Although the RTT value is commonly used to measure the network delay, its value involves the processing delay of the destination host network stack, which distorts the reading accuracy of the network delay. As a remedy, the destination part of SoftFG is responsible for taking the rerouting decision and reading the TCP timestamp of every received packet and subtracts it from the current time to accurately measure the network delay. Indeed, this is possible thanks to the recent advances in NICs [9] that enable nanosecond scale packet time stamping, which boosts the accuracy of timestamp-based network delay measurement.

### A. Load Balancing Policy

For the sake of load balancing and efficient utilization, the SoftFG reroutes large flows of the congested lightpaths onto under-utilized wavelengths, links, or topologies. Common LBs (e.g., [10], [11]) generally offloads the EFs and takes rerouting decision with less care to MFs. However, the size of some MFs resides in a gray area between MFs and EFs. The DC workload distributions present such type of flows that occupies a remarkable share of total DC flows [12], [13]; which are also referred to as cat flows (CFs). At this point, it is worth mentioning that rerouting renders packet reordering problem which is caused by receiving rerouted packets before those packets routed over the old path. The resultant delay caused by this problem is proportional to the flow size [10]. Thereby, SoftFG avoids rerouting MFs to protect them from this undesirable delay. In contrast, the EFs are internally immune from such a problem, where the enhanced performance gains by rerouting mitigate the small loses from packet reordering. Accordingly, SoftFG introduces four types of load balancing:

1) The intra-link load balancing reroutes large flows of a congested wavelength to another under-utilized wavelength in the same FSO link.
2) The inter-link load balancing reroutes large flows between lightpaths routed over different links regardless of their virtual-topology.
3) The intra-topology load balancing reroutes the same class of flows (i.e., CFs or EFs) between lightpaths on the same virtual topology.
4) The inter-topology load balancing reroutes different class of flows (e.g., CFs) between lightpaths on virtual topology of another class of flows (e.g., EFs) .

We must note that the SoftFG does not allow EFs to be rerouted over MF topology as they generally induce performance degradations. In order to avoid routing oscillations and rerouting EF through the MF topology because of the misclassifications (i.e., a detected CF is treated as an EF), the flows are not allowed to be rerouted again from the EF topology back to the MF topology regardless to their class.

### B. Adaptive Lightpath Probing

Short-term LBs require up-to-date statistics of paths to place newly arrived flows or to reroute delayed ones. Periodic path probing is an efficient solution to increase the visibility on network paths and tackle this limitation [11]. However, the proposed fixed period is not sufficient to handle diversified DC workloads, where the less communicating rack-pairs needs different probe interval than the highly communicating ones. Accordingly, we introduce an adaptive lightpath probing method by adjusting the probe interval to the flow arrival-rate. This makes probe intervals proportional to the traffic rate between rack-pairs.

Moreover, the probe interval is upper and lower bounded by thresholds $U$ and $L$, respectively. While $U$ is proportional to the overall DC average flow arrival-rate, $L$ is proportional to the average RTT. These two bounds increase the efficiency of the probing method and reduce the overhead on the high utilized racks. Upon the arrival of a new flow, the probe engine sends a probe message to selected racks and reset the probe interval. However, when the probe interval reaches $L$, and there is no new flow arrives, the probe engine sends a probe to the selected set of racks, as well. On the other hand, if the difference between two flow-arrivals is less than $L$, the probe engine will not send a probe message. The list of selected racks is generated according to the current unexamined paths. For example, if the record has current information about two paths, the probe engine examines the paths other than these two.

### C. Path Update Challenge

The path update challenge appears when a delayed flow is rerouted from a congested path to another one. Provisioned improvement requires multiple RTT to appear on the flow performance. If the LB reads path information table before the improvements appear on the rerouted flow performance, the already rerouted flow will be routed again to another path or go back to its origin.

The native solution to this cumulative challenge is to mark every rerouted flow with a specific flag to avoid future rerouting. Unfortunately, the path utilization levels vary with time, as well as the status of the paths and the rerouted flows. Moreover, the improvements in the performance of the rerouted flow appear in the size of their TCP congestion window size (CWND), and unfortunately, the flows of large CWNDs are highly sensitive to congestion. Thereby, avoiding a large flow from future rerouting degrades the network performance and prolongs the conflict between the congested flows in the over-utilized paths. SoftFG adds two conditions for the newly rerouted flow to be rerouted again:

- A timer proportional to the RTT of average EF CWND,
- A rate comparison between the rate of the rerouted flow before and after the rerouting.

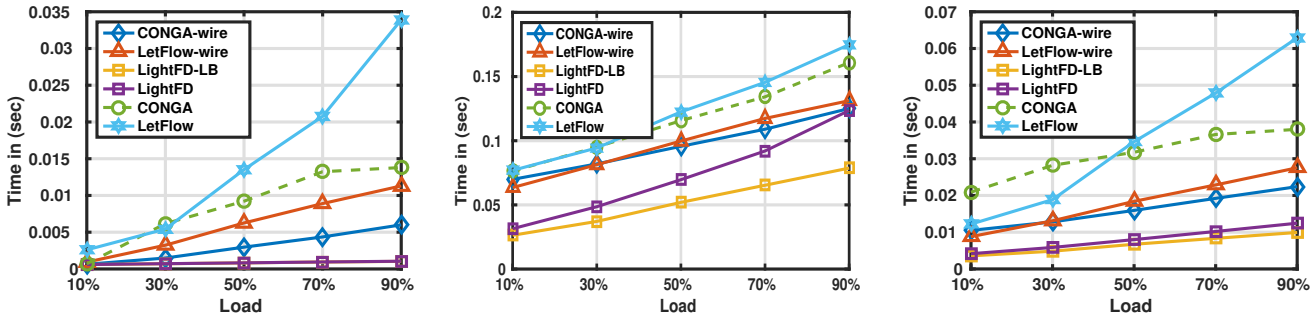
These conditions are necessary to allow enough time to observe the flow performance enhancement. When the timer exceeds a particular threshold value, $T$, and the current flow rate is worse than the previous rate, the flow is rerouted to another path following the path selection conditions or keep it in its current path.

## IV. Evaluations

Using NS3 simulation, we inspect the performance of SoftFG and other schemes under a symmetric $8\times4$ leaf-spine topology with 128 hosts connected by 10Gbps links. We configured the FSO-links with 10 Gbps and the cabled links with 1Gbps, which means FSO links are ten times faster than wired DCN links. The WDM-FSO can be thought of as parallel independent channels. However, they are coupled with the total available lightpath power intensity. Accordingly, in this evaluation part, each FSO link consists of 4 wavelengths. Since the simulation is limited in DCN size, optical channel gains are not distinguishably different due to similar link distances and thus assumed to be identical without loss of generality.

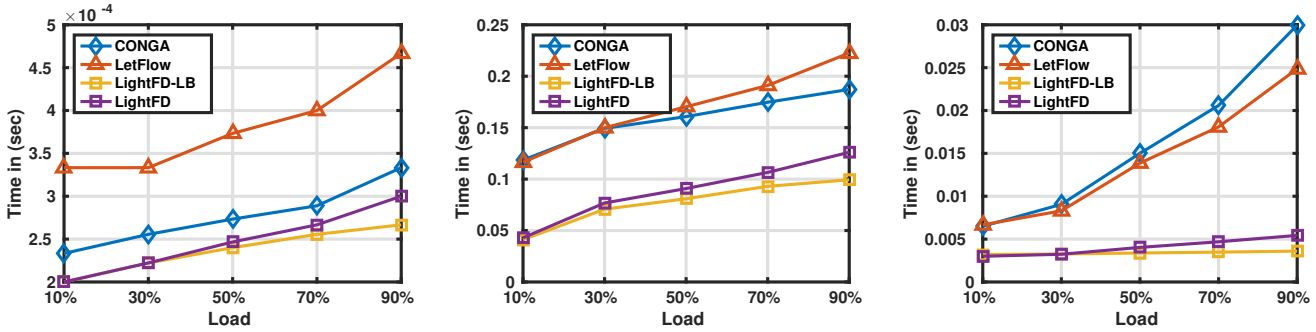
### A. Workloads

Similar to previous works, we perform our simulation using two realistic workloads observed in real DCs: data-mining [12] and web-search [13]. The flow size is generated randomly according to the distributions in the figure. The source and destination servers are randomly selected among the servers in different racks, and the flow between them is randomly generated according to Poisson distribution. The data-mining workloads are more skewed, with about 3.6% of EFs that are responsible for 95% of the exchanged data. Due to these irregular characteristics, the generated topology from the FG-policy for the data-mining workload produces is asymmetric, which has 3Gbps for MFs virtual-topology and 7Gbps for EFs virtual-topology. On the other hand, the topology of the web-search workload is symmetric with 5Gbps for both EF and MF virtual-topologies. The LB task for the asymmetric topology is challenging, where it needs




(a) Comparison of MF-FCTs. (b) Comparison of EF-FCTs. (c) Comparison of overall FCTs.

Fig. 1: The impact of SoftFG on FCT during Web-search workloads.




(a) Comparison of MF-FCTs. (b) Comparison of EF-FCTs. (c) Comparison of overall FCTs.

Fig. 2: The impact of SoftFG on FCT during Data-mining workloads.

accurate and up-to-date statistics in both topologies to perform an efficient load balancing.

### B. Compared Schemes

In the remainder of this section, we compare the performances of the following load balancing schemes:

1) *LetFlow-wire [14]:* is not congestion aware algorithm, it reroutes a flow when a flowlet emerge. We simulate LetFlow according to the parameter settings suggested in [14].
2) *CONGA-wire [15]:* implements congestion aware algorithm on specialized switch chipsets to balance the network traffic load among parallel paths. Similarly, CONGA only reroutes the flow when flowlets emerge. We follow the parameter settings in [15] in CONGA simulation.
3) *LetFlow*: is LetFlow routing algorithm supported by FSO links, i.e., 10× faster then normal DCN. In this routing method, the link capacity is equally divided between the multiple virtual-links, similar to a wavelength in FSO, that is, the capacity of every wavelength, i.e., virtual-link, is fixed to 2.5 Gbps, where 4x2.5Gbps=10Gbps.
4) *CONGA*: is CONGA routing algorithm supported by FSO links and with similar configuration and speed of LetFlow.
5) *SoftFG-LT* is the proposed algorithm employs only the long-term LB.
6) *SoftFG-LB* is the proposed algorithm employs the long-term and short-term LB.

The evaluation in this subsection divided into two parts according to the workload. The SoftFG and other schemes are first evaluated during the web-search workload and then during the data-mining workload. Fig. 1 shows the FCT of MF, EF, and the average FCT of all the flows during the web-search workload. Likewise, Fig. 2 shows the FCT of MF, EF, and the average FCT of all the flows during the data-mining workload. All the figures display the evaluation of SoftFG and other solutions during different traffic loads.

The solutions, i.e., CONGA-wire and LetFlow-wire, in normal wired DCN, show the lowest performance, and their performance is decreasing proportionally to the traffic load. However, CONGA performs better than LetFlow almost in all traffic types, which is expected. SoftFG outperforms CONGA and LetFlow even though they have provisioned with the same link capacities of SoftFG, thanks to the bandwidth efficiency of the proposed algorithm and FG technology. This demonstrates the competence of SoftFG in EF detection and the rapid rerouting of them to their designated topology while leaving the MFs to enjoy the allocated capacity.

The web-search is a high dynamic workload where it demonstrates a high flow arrival-rate and a large number of MFs that yield multiple flowlet gaps. The flow-let LBs prefer

such workloads to exploit the flow-let gaps in balancing the traffic load. During this unstable and dynamic workload, i.e., web-search workload, SoftFG-LB outperforms CONGA and LetFlow solutions by about 3.61x and 7.63x, for MFs and by about 1.63x and 1.7x for EFs, respectively. Also, SoftFG-LB surpasses CONGA and LetFlow, wired solutions, by about 11.2x and 16.5x for MFs and by about 2x and 2.1x for EFs, respectively. At the same workload during the 90% traffic load SoftFG-LB outperforms CONGA and LetFlow solutions by 5.14x and 9.68x for MFs, and by about 1.36x and 1.43x for EFs, respectively. SoftFG-LB surpasses CONGA and LetFlow by about 11.8x and 29x for MFs and by about 1.75x and 1.9x for EFs, respectively.

Similarly, during the data-mining workload and 50% traffic load SoftFG-LB outperforms CONGA and LetFlow by about 1.06x and 1.45x for MFs and by 2.2x and 2.33x for EFs, respectively. In the same context, at the traffic load of 90%, SoftFG-LB outperforms CONGA and LetFlow solutions by about 1.1x and 1.56x for MFs and 1.8x and 2.14x for EFs, respectively. To explain the dissimilarities in output results, the data-mining workload has a large number of EFs, which makes it a bit challenging because the LB encounters a large number of collision incidents between EFs themselves and with MFs. Although the data-mining workload has this challenging characteristic, and there is a small space to improve, SoftFG and SoftFG-LB perform better than other solutions.

## V. Conclusions

In this paper, SoftFG is developed as an agile short-term LB scheme to reroute flows of the congested paths onto underutilized links by means of soft reconfigurations based on the fine-grain network statistics. SoftFG is designed as a kernel module installed on the virtual switches/hypervisor to assure clear visibility on network paths, early congestion detection, and fast-accurate reaction to reroute offloaded flows onto underutilized wavelengths or links. Emulation results show that SoftFG delivers superior performance in comparison with other LB schemes.

## References


[1] A. S. Hamza *et al.*, "Wireless communication in data centers: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1572–1595, thirdquarter 2016.

[2] A. Celik *et al.*, "Wireless data center networks: Advances, challenges, and opportunities," *arXiv preprint arXiv:1811.11717*, 2018. [Online]. Available: https://arxiv.org/abs/1811.11717

[3] N. Hamedazimi *et al.*, "Firefly: A reconfigurable wireless data center fabric using free-space optics," in *Proc. of the ACM SIGCOMM 2014 Conf. on SIGCOMM*, 2014, pp. 319–330.

[4] M. Ghobadi *et al.*, "Projector: Agile reconfigurable data center interconnect," in *Proc. of the ACM SIGCOMM 2013 Conf. on SIGCOMM*. ACM, 2016, pp. 216–229.

[5] A. S. Hamza *et al.*, "Owcell: Optical wireless cellular data center network architecture," in *proc. 2017 IEEE Intl. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[6] A. Celik, A. Al-Ghadhban, B. Shihada, and M.-S. Alouini, "Design and provisioning of optical wireless data center networks: A traffic grooming approach," in *IEEE Wireless Commun. Netw. Conf.(WCNC)*, Apr. 2018, pp. 1–6.

[7] A. Al-Ghadhban, A. Celik, B. Shihada, and M.-S. Alouini, "Lightfd: A lightweight flow detection mechanism for traffic grooming in optical wireless dcns," in *IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[8] A. Celik, A. AlGhadhban, B. Shihada, and M.-S. Alouini, "Design and provision of traffic grooming for optical wireless data center networks," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2245–2259, Mar. 2019.

[9] *Dual Port 10 Gigabit Server Adapter with Precision Time Stamping*. [Online]. Available: https://www.silicom-usa.com/wp-content/uploads/2016/08/PE210G2TSI9-10G-Precision-Time-Stamping-Server-Adapter.pdf

[10] A. Kabbani *et al.*, "Flowbender: Flow-level adaptive routing for improved latency and throughput in datacenter networks," in *Proc. of the 2014 ACM Conf. on Emerg. Net. Exper. and Tech.*, ser. CoNEXT '14, 2014, pp. 149–160.

[11] H. Zhang *et al.*, in *Proc. of the ACM SIGCOMM 2017 Conf. on SIGCOMM*, ser. SIGCOMM '17, 2017, pp. 253–266.

[12] A. Greenberg *et al.*, "Vl2: A scalable and flexible data center network," in *Proc. of the ACM SIGCOMM 2009 Conf. on SIGCOMM*, ser. SIGCOMM '09, 2009, pp. 51–62.

[13] M. Alizadeh *et al.*, "Data center tcp (dctcp)," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. –, Aug. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=2043164.1851192

[14] E. Vanini *et al.*, "Let it flow: Resilient asymmetric load balancing with flowlet switching," in *Proc. of the 14th USENIX Conf. on Net. Syst. Desi. and Implem (NSDI).*, ser. NSDI'17. Berkeley, CA, USA: USENIX Association, 2017, pp. 407–420. [Online]. Available: http://dl.acm.org/citation.cfm?id=3154630.3154664

[15] M. Alizadeh *et al.*, "Conga: Distributed congestion-aware load balancing for datacenters," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 503–514, Aug. 2014.