An INT-based Load Balancing Mechanism for Cloud Datacenters

Zixi Cui*, Yuxiang Hu and Saifeng Hou
Information Engineering University, Zhengzhou, China
*Corresponding author: cuizixi@zkya.com

Abstract—In this work, we propose Closer, a scalable congestion-aware load balancer for cloud datacenters. Closer complies with the evaluation of technology including the deployment of regular Clos topologies, overlays for network virtualization, and virtual machine (VM) clusters. Leveraging In-band Network Telemetry (INT) to obtain precise link load information, we design a simple but efficient algorithm that implements weighted ECMP at the edge of fabric, which enables Closer to proactively map the flows to the appropriate path and avoid the excessive congestion of a single link. Closer decouples the system into centralized route calculation and distributed route decision to guarantee its efficiency and stability in large-scale networks. In tested experiments, Closer achieves 2–7 times better flow completion time (FCT) than existing alternatives at 70% network load.

1. INTRODUCTION
To achieve high-throughput and low-latency communication between server clusters, the switch-oriented topologies (e.g., Leaf-Spine and Fat-Tree) are widely used for the physical carrier network of datacenters [1]. These topologies follow the Clos architecture that aims to make full use of the switch’s performance. Typically datacenters use Equal-Cost Multi-Path (ECMP) algorithm for load balancing to adapt to the network characteristic of multipath. ECMP randomly selects one from available paths to the data flows, making it highly susceptible to excessive congestion of local links caused by hash collisions of large flows [2, 3].

Although a variety of works have been proposed from several different perspectives to address the shortcomings of ECMP, both of them come with significant deployment challenges and performance limitations that largely prevent their adoption. Centralized schemes have been proved to be too slow for the volatile traffic patterns in datacenters [4]. Host-based methods such as MPTCP [5] split each connection into multiple sub-flows by rebuilding the kernel network stack, and hence, actually, increase the burstiness of traffic and bring an additional burden to the end-host. Network-layer protocols (e.g., CONGA [4] and HULA [6]) recursively implement link state awareness and route calculation in the data plane. Such radical revolutionary schemes require replacing every network switch (physical or virtual) with one that implements a new state-propagation and load-balancing algorithm [7].

Can we implement the dynamic traffic scheduling in a patching manner, so that it can be deployed quickly with the infrastructure of the ECMP-based network but overcomes the defects of the congestion-oblivious method? We believe that the source routing at the edge of fabric provides a unique opportunity to achieve this goal. This has been done in Clove [7], where the virtual switch in the hypervisor would direct the traffic over other less congested paths if they receive Explicit Congestion Notification (ECN) signal from an exhausted switch. However, Clove is not stable and configurable in
large-scale networks as it offloads entire handling logic into software switches, without any participation of the controller. Continuing their research course, we propose Closer, a scalable INT-based load-balancing framework for cloud datacenters, decoupling the system into centralized route calculation and distributed route decision.

Our major contributions are summarized as follows:

- We design a load-balancing application framework that entirely complies with the existing technical route of cloud datacenters and takes advantage of Software-Defined Networking (SDN).
- We propose Closer, a traffic scheduling algorithm over multi-path. Closer realizes real-time feedback loops with INT and efficiently load balancing with weighed-ECMP.
- We implement a Closer prototype by developing open software projects. The experimental results show that it outperforms the counterparts in average FCT.

2. THE CHALLENGE AND OPPORTUNITY FOR CLOSER

2.1 Challenge
Clos architecture is highly normalized and homogeneous. In a 2-tier Clos topology, there are hundreds of high-bandwidth links across the spine-tier between every pair of leaf switches, providing physical prerequisites for large amounts of traffic forwarding. Under the condition that both the network topology and the network traffic are symmetrical, ECMP should approach near-optimal load balancing within fabric [8]. However, the symmetry of the model is likely to be destroyed because (i) the switches are heterogeneous as they are from several vendors; (ii) the shutdown of nodes/links occurs frequently; (iii) the traffic distribution is uneven (more than 80% of VMs have never had communication, and 98% of VMs have throughput of less than 20kbps). Static routing policy can no longer meet the QoS for tenants. How to detect network failures timely to reroute traffic? How to obtain network-wide link status information for optimal load-balancing algorithm? These issues are huge challenges for cloud datacenters.

2.2 Opportunity
With programmable data plane technology that enables fast reconfigurable packet-processing pipelines, INT brings great breakthroughs to the development of network measurement [9]. Unlike passive measurement methods (e.g., sFlow and Net-Flow), INT provides data plane services to actively collect, deliver, and report network state. Specifically, INT generates a probe packet (special packet or modified normal packet) as the messenger of query instruction and injects them into the network. The INT-capable node will directly export device-internal states to the telemetry monitoring system, or push them into the metadata stack of probe packet as required. This “packet-as-instruction” framework not only provides a new mode of operation to achieve end-to-end traffic management and network event detection but also brings a new opportunity to achieve real-time control loops for route decision.

3. CLOSER DESIGN
This section elaborates the design of INT-based congestion-aware load balancing algorithm and provides the example of Closer’s deployment over the VXLAN network.
3.1 INT for Overlay Network
In a multi-tenant cloud datacenter, overlay gateways are responsible for encapsulating packets received from a VM so that L2 frames can be transmitted across the IP network. The most popular network virtualization tunneling protocol is VXLAN, which is known as the cornerstone of cloud computing. As technology evolves, the IETF has been drafting the VXLAN GPE (Generic Protocol Extension for VXLAN) [10] standard so that the inner packet is no longer limited to Ethernet frame.

In the INT dataplane specification v2.1 [11], released by the P4.org Applications Working Group in June 2020, the encapsulation format for INT over VXLAN GPE is defined (see Fig. 1). Closer uses the most classic Hop-by-Hop mode of operation (a.k.a. “INT-MD” mode) for path monitoring and link awareness. The vSwitch at the source hypervisor embeds the INT headers into the packets that sampled from the active flow at a fixed frequency. In each device along the forwarding path, the probe packet will extract the local node id and Tx utilization of the egress interface in turn. When the probe traverses the network to the other end, the destination hypervisor is supposed to extract the INT headers and report the path state to the remote controller. Sampling frequency is approximately set to several hundred milliseconds to limit the bandwidth overhead.

3.2 Real-time Feedback loop
In the environment of SDN, the controller is responsible for monitoring the operating status of the entire network service. In order to avoid the dilemma of local routing decisions, all collected network state information should be analyzed centrally before being fed back to the data plane. The entire working framework is shown in the Fig. 2. The network state database is maintained by controller. After simple data classification, the controller can realize the exhaustive congestion information between any two host-end, and provide material for the next path-selection stage. Closer chooses the utilization of the bottleneck link as the only criterion for evaluating the quality of an end-to-end path—the lower the link utilization of the bottleneck link, the better the whole path. Due to the limited memory resources for software switches, the controller only selects several items from the hundreds of available paths to ensure that the table entries would not overwhelm the software switch.

![Figure 2. INT-based network feedback loop.](image-url)

Closer adopts the strategy of “pre-allocation and dynamic-tuning” to make the system efficient. During the initial phase, the controller picks k path that shares the least number of links each other for every pair of overlay tunnel. A simple greedy algorithm is accepted as there is no traffic load at the moment. After that, the weight of paths varies with the fluctuation of the bandwidth utilization of the bottleneck link. The controller is also update entries in time to ensure that the source vSwitch always maintains the k-optimal paths for every destination. Note that this process may involve the competition that high-traffic nodes preempt the pre-allocated bandwidth of low-traffic nodes.
3.3 Indirect Source Routing Decision for Flowlet

In order to implement ECMP-based load balancing simply and efficiently, each switch in the overlay network generally distinguish different flows based on the 5-tuples fields of the encapsulation header, and use the hash-threshold method to average them to all next hops. In particular, four elements of the 5-tuples—the source and destination IP addresses, the transport protocol, and its destination port number—have been fixed before communication (UDP and port 4789 for VXLAN by default). Therefore, there is a clear mapping relationship between the source port number and the route the packet takes through [7] the network. Given this feature, we build a linkage of path selection and packet encapsulation: the tables from the controller record the source port and weight corresponding to a selected path. The vSwitch selects appropriate source port according to probability to control congestion, which implements indirect source routing at the entrance of the overlay network.

Next, before using the weighted round robin (WRR) for route decision, we also consider relieving the inefficiencies of uneven traffic distribution on equal-cost paths by dividing long-lived flows into small units and routing these units independently. Based on previous work [3, 4, 6], Closer splits the flow into flowlet by an internal threshold to make sure that packets are handled with the updated route but received in order as much as possible. In theory, the flowlet inter-packet gap more than the maximum round-trip time (RTT) can eliminate the reorder delay of TCP messages caused by the different forwarding paths. The whole routing mechanism is shown in Fig. 3.

![Figure 3. W-ECMP routing in Closer.](image)

4. EVALUATION

We build a virtual network environment within emulator Mininet and develop a basic prototype to demonstrate the proposed Closer. The hardware is with Intel Xeon silver 4114 CPU and 64GB memory, running Ubuntu 18.04 OS. To evaluate the performance of Closer, we set up ECMP, MPTCP and Clove as control groups in the same environment.

4.1 Experimental Setting

**Testbed.** As shown in Fig. 4, the 2-tier Clos topology in our testbed consists of four spine switches (S1–S4), four leaf switches (L1–L4). Each leaf is connected to five servers that respectively contain an access virtual switch connected to a VM. There are a total of four disjoint paths that traffic could take from one leaf to another. The core of the network builds a VXLAN fabric using the modified OVS, while the access switch in the hypervisor is a P4-enabled software switch (bmv_2). All of the switches are uniformly managed by an ONOS controller (which supports both OpenFlow and the P4Runtime). Due to the limited performance of the software switch, the link bandwidth between OVS is set to 1G, and the servers gain access to the network with 200M links so that the core-tier is not oversubscribed.

**Traffic workload.** We use two widely accepted datacenter traffic models, web-search and data-mining, in testbed experiments. Both of these workloads are heavy-tail for the cumulative distribution of flow sizes. The difference is that 80% of Data-mining traffic is less than 10KB in size, while the
most Web-search traffic is concentrated between 10KB and 10MB. We scale the dimensions of both traffics appropriately to accommodate the network bandwidth.

![2-tier Clos topology used in experiments.](image)

Figure 4. 2-tier Clos topology used in experiments.

Powered by Iperf3 utility, we develop a client-server script, where half of the endpoints work as clients and randomly establish a persistent TCP connection to the servers under the different leaf. As a result, there are total of 10 VNI tunnels in the overlay network. The generation of flows is according to a Poisson process, and we change the mean of the inter-arrival rate to achieve the desired network load (10% – 90%). Ten random seeds are running for each experiment and we measure the average FCT of the ten runs as the main performance metric. All experimental values are normalized to the results of ECMP.

4.2 Baseline Symmetric Topology

Fig. 5 shows the result of four algorithms for the two workloads with the baseline topology. Other than MPTCP, the overall average FCT is similar to each other for all schemes, especially in the Web-search workload. When the network load continues to remain low, the simple deployment of ECMP approach near-optimal load balancing, while the complex control loop would slightly reduce the efficiency of the whole system (the processing and spread delay for congestion signal). However, the advanced nature of the congestion-aware scheme begins to manifest as the load increases up to 70%. Clove and Closer gradually narrow the gap with ECMP and realizes the reverse overtaking. As for MPTCP, 4 sub-flows per connection could accelerate the time the traffic spent in the network, but it also increases congestion at the edge links due to the burstiness of multiple sub-flows.

![Web-search overall avg FCT](image)

(a) Web-search overall avg FCT
From the distinction between Fig. 5(a) and Fig. 5(b), we observe that ECMP prefers web-search and does quite well, which results from the relatively even distribution of flow size. On the contrary, the data-mining is more “heavy”—there are more hash collisions of large flows that seriously damage the performance of ECMP. Clove and Closer are resilient to flow distribution, achieving more than 50% better FCT at 90% load.

4.3 Asymmetric Topology
Simulating asymmetry in the baseline symmetric topology, we set up the link between S4 and L4 with 500M. Correspondingly, the scope of the statistical object is adjusted to all web-search traffic through the switch L4. At this point, the total available bandwidth is reduced by 12.5%, and thus the traffic loads should be balanced carefully to prevent the bottleneck link (S4-L4) from being congested.
Fig. 6 exposes the drawback of the congestion-oblivious method. As the offered load increases beyond 50%, the whole system is in an embarrassing state that the bottleneck link is sharply congested while the remaining links still have a large amount of available bandwidth, which makes ECMP’s performance drastically deteriorate. Closer is superior to MPTCP and Clove in terms of sensitivity and performance. This is because Closer proactively adjusts traffic before congestion occurs and maintain a proportional increase of load across all links. This makes Closer achieve 6.7x better performance than ECMP and 1.4x better than Clove at 70% network load.

Fig. 6(b) shows the 99th percentile FCT of all schemes. One of the most notable is the change of MPTCP. Although MPTCP performs better in terms of average FCT in asymmetric topology, it is significantly worse than Clove or Closer on 95th FCT; i.e., MPTCP suffers from long-tail latency. As described in [7], the consistency of “flow” is destroyed in MPTCP, where there is at least a sub-flow of each connection get mapped to the congested path. Closer still maintains the best performance metrics, which achieve 3x better tail latency compared to MPTCP and 2x better to Clove at 60% network load.

5. CONCLUSION
This paper presents the design, implementation and evaluation of Closer, a scalable load-balancing algorithm for cloud datacenters. In the control plane, Closer would collect network global status information with In-band telemetry technology in real-time. In the data plane, the modified software switch can obtain the necessary information for routing decision from the controller and execute the traffic scheduling algorithm independently. This proactive approach achieves a good load balancing performance, which illustrates the advantages of SDN architecture. The experimental results show that Closer outperforms the counterparts both in average and 99th FCT.

ACKNOWLEDGMENT
We thank the anonymous reviewers for their insightful feedback. This work was supported in part by National Key R&D Project of China under grant 2019YFB1802501.

REFERENCES
[1] M. Al-Fares, A. Loukissas, and A. Vahdat. “A scalable, commodity data center network architecture.” Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, vol. 38, no. 4, 2008, pp. 63–74.
[2] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. “Hedera: dynamic flow scheduling for data center networks.” NSDI’10 Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, 2010, pp. 19–19.
[3] S. Kandula, D. Katabi, S. Sinha, and A. Berger. “Dynamic load balancing without packet reordering.” Acm Special Interest Group on Data Communication, vol. 37, no. 2, 2007, pp. 51–62.
[4] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, … G. Varghese. “CONGA: distributed congestion-aware load balancing for datacenters.” Acm Special Interest Group on Data Communication, vol. 44, no. 4, 2015, pp. 503–514.
[5] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. “Design, implementation and evaluation of congestion control for multipath TCP.” NSDI’11 Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, 2011, pp. 99–112.
[6] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford. “HULA: scalable load balancing using programmable data planes.” Proceedings of the Symposium on SDN Research, 2016, p. 10.
[7] N. Katta, A. Ghag, M. Hira, I. Keslassy, A. Bergman, C. Kim, and J. Rexford. “Clove: congestion-aware load balancing at the virtual edge.” Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies, 2017, pp. 323–335.
[8] K. He, E. Rozner, K. Agarwal, W. Felter, J. Carter, and A. Akella. “Presto: edge-based load balancing for fast datacenter networks.” Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, vol. 45, no. 4, 2015, pp. 465–478.
[9] T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, …Y. Liu. “INT-Path: towards optimal path planning for in-band network-wide telemetry.” IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 487–495.

[10] “Generic Protocol Extension for VXLAN.” https://tools.ietf.org/html/draft-ietf-nvo3-vxlan-gpe-09, 2019.

[11] “In-band Network Telemetry (INT) Dataplane Specification v2.1.” https://github.com/p4lang/p4-applications/blob/master/docs(INT_v2_1).pdf, 2020.