EXTENDED-ABSTRACT

CLEO: Machine Learning for ECMP

HEESANG JIN, Korea University, Seoul, South Korea

MINKOO KANG, Korea University, Seoul, South Korea

GYEONGSIK YANG, Korea University, Seoul, South Korea

CHUCK YOO, Korea University, Seoul, South Korea

Published: 09 December 2019

CoNEXT '19: The 15th International Conference on emerging Networking EXperiments and Technologies
December 9 - 12, 2019
FL, Orlando, USA

Conference Sponsors:
SIGCOMM

https://doi.org/10.1145/3360468.3366768
ISBN: 9781450370066

CLEO: Machine Learning for ECMP

Heesang Jin
Korea University
hsjin@os.korea.ac.kr

Minkoo Kang
Korea University
mkkang@os.korea.ac.kr

Gyeongsik Yang
Korea University
ksyang@os.korea.ac.kr

Chuck Yoo
Korea University
chuckyoo@os.korea.ac.kr

ABSTRACT

In this paper, we propose CLEO, which is a machine learning approach to equal-cost multipath routing (ECMP) schemes to distribute and balance traffic. ECMP-based traffic load-balancing is widely practiced by datacenters, but hash collision resulting from skewed ECMP hashing makes it difficult to achieve the desired throughputs over paths. Various solutions have been proposed to overcome the performance degradation caused by hash collision, but most of these solutions require modifying packet headers or replacing switches. To solve this problem, CLEO builds a neural-network model that characterizes the ECMP scheme of a switch. The proof-of-concept evaluation shows that CLEO improves the root mean square error fourfold between the desired and real path throughputs.

CCS CONCEPTS

• **Networks** → *Data path algorithms*; *Network management*; *Network performance analysis.*

ACM Reference Format:
Heesang Jin, Minkoo Kang, Gyeongsik Yang, and Chuck Yoo. 2019. CLEO: Machine Learning for ECMP. In *The 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '19 Companion), December 9–12, 2019, Orlando, FL, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3360468.3366768

1 INTRODUCTION

Most datacenter network topologies are symmetric such that multipaths exist between any two hosts. All paths are assumed to have an equal cost in terms of packet forwarding [6]. Based on this assumption, most datacenters use a traffic load-balancing (LB) technique that distributes traffic on multipaths to enhance link utilization and network throughput. The most popular LB is equal-cost multipath routing (ECMP), which uses packet header hashing. Figure 1 depicts ECMP, which consists of two primary processes: 1) hashing and 2) path selection.

ECMP hashes the given packet header fields such as five-tuples of the header (one key per flow). With the generated hash key, the path selection process checks candidate paths that are provided by the routing element, such as routing protocols within a switch or external controllers in software-defined networking. ECMP then selects one path in a round-robin manner or by statically mapping the hash keys and paths. ECMP also works with switches that receive a path ratio [7]. The path ratio is given for a switch to achieve a "desired" throughput for paths from the LB perspective. For example, if three paths receive a 1:2:3 ratio, then the switch transmits 1/6, 2/6, and 3/6 traffic to the paths, respectively. The path selection process of ECMP then follows the ratio over three paths.

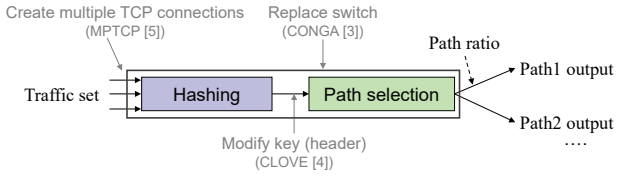


Figure 1: ECMP mechanism and previous approaches to solve ECMP hash collisions.

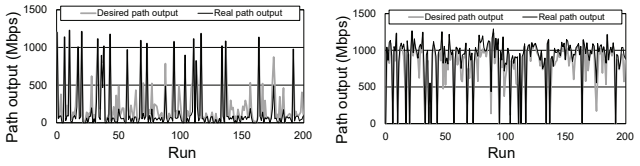
ECMP is efficient in distributing traffic with O(1) time complexity because it is based on hashing. However, it has a critical problem: hash collision during the hashing process. Hash collision occurs if the packet header fields do not generate evenly distributed keys. When the generated keys are not evenly distributed, the path selection process cannot follow the path ratio accurately. This means that even though ECMP follows the path ratio to distribute the traffic, it cannot achieve the desired throughput for paths.

Furthermore, although ECMP is supported by various switches in datacenters, ECMP schemes are different at various switches because ECMP has no standard implementation. For example, the packet header fields to be hashed are different: five fields (source, destination IP addresses, IP protocol field, source, and destination port) [7], six fields (in which an ingress port is added to the previous five fields) [1], and nine fields (source, destination Ethernet addresses, etherType, VLAN ID, source, destination IP addresses, IP protocol, source, and destination port) [2]. In addition, the path selection process with the path ratio differs depending on the switch (e.g., multiplying the path ratio over the hashed key [2] and weighted round-robin selection considering the path ratio [1]). It is widely acknowledged that realizing the desired path output using ECMP is challenging. Although various solutions (e.g., MPTCP [5], CLOVE [4], and CONGA [3]) have been proposed, they require modifications to packet headers or the replacement of switches.

As an alternative, this paper proposes CLEO, which learns the characteristics of an ECMP scheme. CLEO trains a neural-network model based on the traffic set and path output from the switch. When training is complete, the model outputs a path ratio that can realize the desired path output. CLEO builds its model for a specific switch so that it can be applied to different ECMP schemes through training. Regarding the accuracy of the real path output over the desired path output, our evaluation results show that CLEO

CoNEXT '19 Companion, December 9–12,2019, Orlando, FL, USA

ACM ISBN 978-1-4503-7006-6/19/12.
https://doi.org/10.1145/3360468.3366768



(a) First path. **(b) Second path.**

Figure 2: Path output differences.

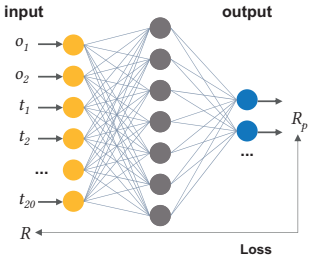


Figure 3: CLEO model.

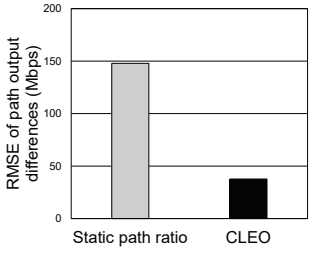


Figure 4: RMSE of path output differences.

increases accuracy by four times as much as that of the ECMP that follows the path ratio statically.

2 MOTIVATION

We conduct experiments based on Open vSwitch that implements ECMP. We create a topology composed of two hosts and two paths between them. The experiments consist of 200 runs, where each run creates 20 TCP connections. The throughput of the individual connection is randomly chosen between 10 and 100 Mbps with iperf3, and the path ratio for the two paths ($r1 : r2$) is given randomly ($r1$ and $r2$ between 0 and 10). Assuming that the total throughput of 20 connections is 2,000 Mbps and that the path ratio of the two paths statically calculated is 1:4, the desired path output of the first and second paths would be 400 and 1,600 Mbps, respectively.

To view the path output in the network (real path output), we generate similar traffic and set the path ratio to Open vSwitch. Figure 2 illustrates the path outputs measured for 200 runs per path. The x-axis represents the individual run, and the y-axis shows the desired path output (in gray) and real path output (in black). We consider each run a "success" if the difference between the desired and real path output is less than 5% of the desired path output. The results show that only 36% of 200 runs are successful. We find that skewed keys are the main cause of the results.

3 DESIGN AND EVALUATION

Figure 3 depicts the CLEO model. The model takes a traffic set and desired path output pair as its inputs and then predicts a path ratio to realize the desired path output pair. To build the model, we conduct two steps: 1) sample data generation and 2) model training.

First, CLEO generates a sample dataset (D) for use in model creation as follows. 10K data (D= $\{d_1, d_2, \ldots, d_{10,000}\}$) are generated by running iperf3 with the same switch (Open vSwitch) and a topology as described in the previous section. d_i consists of a traffic set (T= $\{t_1, t_2, \ldots, t_{20}\}$), path ratio ($R$= $\{r1 : r2\}$), and path output pair (O= $\{o1, o2\}$). For example, we get path output pair as 50 and 150 Mbps when each network connection is 10 Mbps and the path ratio to the Open vSwitch is 1:2.

To generate sample data (d_i), CLEO randomly selects the values of T between 10 and 100 Mbps and R between 0 and 10. The reason for this random selection is that each d_i is processed by ECMP independently, which can be considered as $i.i.d$. The selected T and R are entered into the Open vSwitch and O is measured.

Next, 7K of 10K sample data are used to train the CLEO model. The model is built as a multiclass neural network that formulates this problem as a multinomial classification problem. During training, we feed a traffic set (T) and path output pair (O) as input, and a path ratio (R) is used as the label for the T and O. Then, the CLEO model is iteratively updated to generate the predicted R (R_p) for the given T and O. It calculates the loss between R_p and label R using cross entropy and updates the model to reduce the loss.

Second, to evaluate the accuracy of the CLEO model, CLEO uses the remaining 3K data from the randomly generated traffic set. The evaluation is conducted as follows. We first feed T and O into the CLEO model and get R_p. The R_p is the predicted value from the CLEO model that satisfies the O for the T. Next, CLEO sets the T and R_p on the Open vSwitch. Then, a real path output pair (O_r) is measured. For 3K data, we perform this process repeatedly and calculate the root mean square error (RMSE) between O and O_r.

Figure 4 shows the RMSEs for the ECMP that follows the path ratio statically (static path ratio) and CLEO. Note that the traffic set and desired path output of each run are the same in Figure 4, but the real path output is measured both by the static path ratio and CLEO. Results show that CLEO reduces the RMSE by 75% (fourfold improvement).

4 CONCLUSION AND FUTURE WORK

In this paper, we present the initial design and proof-of-concept for CLEO, a machine learning-based ECMP. The evaluation results show that CLEO improves accuracy up to four times. Because CLEO does not modify packet headers nor replace switch hardware, CLEO can be used for various ECMP switches and dynamic traffic. In the future, we plan to enhance the accuracy of CLEO by applying a sophisticated machine learning model, extending the input data, and tuning the model parameters. We will also apply CLEO to more complex networks with a greater number of paths.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful comments. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2015-0-00288, Research of Network Virtualization Platform and Service for SDN 2.0 Realization, and No. 2015-0-00280, (SW Starlab) Next generation cloud infra-software toward the guarantee of performance and security SLA).

REFERENCES

[1] 2019. Equal Cost Multipath Load Sharing - Hardware ECMP | Cumulus Linux 3.7. https://docs.cumulusnetworks.com/cumulus-linux/Layer-3/Equal-Cost-Multipath-Load-Sharing-Hardware-ECMP/. (2019). (Accessed on 09/21/2019).

[2] 2019. www.openvswitch.org/support/dist-docs/ovs-ofctl.8.txt. http://www.openvswitch.org/support/dist-docs/ovs-ofctl.8.txt. (2019). (Accessed on 09/20/2019).

[3] Mohammad Alizadeh, Tom Edsall, Sarang Dharmapurikar, Ramanan Vaidyanathan, Kevin Chu, Andy Fingerhut, Francis Matus, Rong Pan, Navindra Yadav, George Varghese, et al. 2014. CONGA: Distributed congestion-aware load balancing for datacenters. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 503–514.

[4] Naga Katta, Aditi Ghag, Mukesh Hira, Isaac Keslassy, Aran Bergman, Changhoon Kim, and Jennifer Rexford. 2017. Clove: Congestion-Aware Load Balancing at the Virtual Edge. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM, 323–335. https://doi.org/10.1145/3143361.3143401

[5] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. 2011. Improving datacenter performance and robustness with multipath TCP. In *ACM SIGCOMM Computer Communication Review*, Vol. 41. 266. https://doi.org/10.1145/2043164.2018467

[6] Jiao Zhang, F. Richard Yu, Shuo Wang, Tao Huang, Zengyi Liu, and Yunjie Liu. 2018. Load balancing in data center networks: A survey. *IEEE Communications Surveys and Tutorials* 20, 3 (2018), 2324–2325. https://doi.org/10.1109/COMST.2018.2816042

[7] Junlan Zhou, Malveeka Tewari, Min Zhu, Abdul Kabbani, Leon Poutievski, Arjun Singh, and Amin Vahdat. 2014. WCMP: Weighted cost multipathing for improved fairness in data centers. In *Proceedings of the Ninth European Conference on Computer Systems*. ACM, 5.