SDN-Based Dynamic Multipath Forwarding for Inter-Data Center Networking

Yao-Chun Wang, Ying-Dar Lin
Dept. of Computer Science
National Chiao Tung University
Hsinchu, Taiwan

Guey-Yun Chang
Dept. of Computer Science
National Central University
Taoyuan, Taiwan

***Abstract*—Since traffic engineering (TE) in Software Defined Networking (SDN) can be much more efficiently and intelligently implemented, Multipath in SDN becomes a new option. However, Ternary Content Addressable Memory (TCAM) size become the bottleneck of SDN. In this paper, we propose an SDN-based Dynamic Flowentry-Saving Multipath (DFSM) mechanism for inter-DC WAN traffic forwarding. DFSM adopts source-destination-based multipath forwarding and latency-aware flow-based traffic splitting to save flow entries and achieve better load balancing. Our evaluations indicate that DFSM saves 15% to 30% system flow entries in different topologies compared to label-based tunneling, and also reduces average latency by 10% to 48% by consuming 8% to 20% more flow entries than Equal-Cost Multipath (ECMP) in less-interconnected topologies. In addition, compared to even traffic splitting, DFSM reduces the standard deviation of path latencies from 14% to 7%.**

Keywords—SDN, OpenFlow, Inter data center, Multipath

I. INTRODUCTION

ISP today builds clouds to provide various on-line services. In recent trends, a cloud often consist of multiple datacenters (DC) located in different geographic areas. For inter-DC networks, a large scale cloud often uses wide area network (WAN) to connect multiple DCs.

Many services in cloud rely on low-latency to enhance user experience. Due to the large amounts of rapid growth inter-DC traffic, traffic engineering (TE) mechanism is often used in inter-DC WAN to improve network performance. TE methods based on Equal-Cost Multipath (ECMP) [1] are widely used in intra-DC networks to enhance the performance of horizontal transmission. However, the adoption of equal-cost shortest paths may result in an inability to fully utilize link resources in irregular topologies, the performance improvement by applying ECMP on inter-DC WAN may be limited.

Compared to IP/MPLS-based TE in traditional networks, TE in SDN [2] can be much more efficiently and intelligently implemented as a consequence of the unified global view of complicated networks and flexible traffic control ability. Both Google [6] and Microsoft [7] adopt SDN in their inter-DC WAN, using multiple tunnel paths with traffic sharing for each edge DC pair to maximize the utilization of network links.

OpenFlow [3] is a popular SDN standard. Under the OpenFlow environment, switches rely on large Ternary Content Addressable Memory (TCAM) to save flow entries (forwarding rules), which results in TCAM size becoming the bottleneck of SDN. To support ever-increasing bandwidth demands and service expectations, CORD [4] re-architects the Telco Central Office as a datacenter, combining SDN, Network Functions Virtualization (NFV), and elastic cloud services to build cost-effective and agile networks. As a result of this trend, the number of DCs will explode in future years (for example, AT&T currently operates 4700 Central Offices). For SDN-based inter-DC WAN, the flow entries increase as the number of DCs increase, and consequently, scalability should be considered.

In this paper, we propose a Dynamic Flowentry-Saving Multipath (DFSM) mechanism for traffic forwarding in SDN-based inter-DC WAN, to satisfy the latency demand of each DC pair with fewest flow entries and lowest standard deviation of path latencies.

II. PROPOSED METHODS

A. Overview

DFSM periodically adjusts multiple paths used by each existing DC pair in order to adapt traffic change and satisfy their latency demands with the fewest flow entries and the lowest standard deviation of path latencies.

To recognize whether a DC pair reaches its latency demand, a demand fulfillment status (with three possible values: *over-satisfied*, *satisfied* and *unsatisfied*) is recorded for each DC pair. The idea behind dynamic path adjustment is to release paths for over-satisfied DC pairs, and to allocate paths for unsatisfied DC pairs. Furthermore, the fairness between DC pairs is considered. Fairness for DC pairs means the reduction of differences between the demand satisfaction degrees of DC pairs. The satisfaction degree of a DC pair (s_s, s_d) is defined as demanded latency divided by actual average path latency.

The dynamic path adjustment of DFSM consists of the following three components, and the details of each component is dealt with in following subsections.

1) Flow-entry-saving multipath: DFSM assigns least-latency paths to each DC pair in order to meet latency demand with fewer paths, and also uses source-destination-based forwarding to forward packets to the possible next hops, based on both its source and destination. Compared to popular label-based tunneling [7] (i.e. the traffic is split and labeled at

978-1-5386-0728-2/17/$31.00 ©2017 IEEE

source node, to be forwarded along each tunnel path to the destination based on its assigned label), source-destination-based forwarding merges flow entries at intersection forwarding nodes (or split points) crossed by multiple paths.

2) Latency-aware traffic splitting: Traffic splitting can occur at every intersection forwarding node under source-destination-based forwarding. source-destination-based forwarding adopted in DFSM is based on latency-aware flow-based traffic splitting, which can achieve better path load balancing. DFSM aims to offer equal latency paths for each DC pair, so as to prevent flows of the same DC pair from experiencing widely different latency due to the commonly used hash-based path selection.

3) Performance assurance: in order to check whether an attempted path addition (or removal) for a DC pair increases (or decreases) its satisfaction and increases fairness, DFSM adopts a performance prediction procedure to decide whether the attempted path adjustment should be confirmed.

B. Flow-Entry-Saving Multipath

1) Greedy path selection: In order to meet latency demand and low TCAM requirements, DFSM chooses the least latency paths. DFSM computes *k*-least-hop-count available paths for each newly-added DC pair (*k*-least-hop-count available paths for an existing DC pair is determined when it is newly-added). During periodic path adjustments, DFSM determines the latency of these pre-computed available paths, and assigns the least-latency paths to unsatisfied DC pairs.

2) Source-destination-based forwarding: Compared to label-based tunneling [7], source-destination-based forwarding can merge per-tunnel flow entries on intersection forwarding nodes (or split points) crossed by multiple paths (i.e. for each pair, each split point needs only 1 flow entry and at most 1 group entry to split flows to multiple next hops), therefore saving flow entries.

C. Latency-Aware Traffic Splitting

DFSM aims to offer equal latency paths for each DC pair, by deciding split ratios at each split point based on the consideration of path latency, so as to balance the loading (or latency) of paths assigned to each DC pair. In the literature [5] [7], the basic idea of path load balancing is to have the split ratio become inversely proportional to the path load. DFSM adopts the same idea, but different to the split ratio decision, only at source node [5] [7] and only for edge-disjoint paths in the literature [5]; the split ratio decision of DFSM fits into our least latency path (whose edges could be non-disjoint) selection to achieve better traffic control. For a split point in DFSM, the split ratio corresponding to a designated next hop is inversely proportional to the average latency of the partial paths that starts with the split point and proceeds via the designated next hop to the destination.

D. Performance Assurance

Obviously, there are many path additions/removals in dynamic path adjustment. To achieve better performance, DFSM estimates the network link latencies caused by a path addition or removal by the prediction procedure. The prediction procedure first rebalances traffic load after a path addition or removal, i.e. re-computes the ideal distribution of the ingress load of the DC pair (i.e. the total traffic load send from a DC to another DC). In order to reallocate the traffic load of a given DC pair, e.g., *P*, we roll back to an environment with all the traffic loads of existing DC pairs (except the given DC pair *P*), and re-assign the traffic load of *P* to paths for it. The prediction procedure then estimates the latency of each link with traffic load, link bandwidth, and M/M/1 traffic model. With the estimated network link latencies, DFSM can then compute the average path latency for the DC pair, and check whether the latency demand has been fulfilled.

E. Dynamic Path Adjustment

To achieve fairness for DC pairs, DFSM allocates available path resources to each unsatisfied DC pair during the path adjustment process, and executes path adjustment process again and again until all DC pairs are satisfied or nothing can be improved (no available paths remain). During an adjustment process, DFSM iteratively picks the lowest-satisfaction DC pair *P* among all unsatisfied DC pairs, and allocates the least-latency path to *P*.

To evaluate the degree of satisfaction of all DC pairs after path allocation to a DC pair (s_s, s_d), the prediction procedure in DFSM not only re-determines the split ratios of the allocated paths for pair (s_s, s_d), but also re-estimates the latencies of links of these allocated paths. Using the most up-to-date link latencies, DFSM can evaluate the satisfaction degree of all DC pairs. Note that an attempted path addition for a DC pair will be confirmed only when the predicted satisfaction degree of the DC pair increases and the standard deviation of the predicted satisfaction degree of all DC pairs decreases. The dynamic path adjustment is illustrated in Fig. 1.

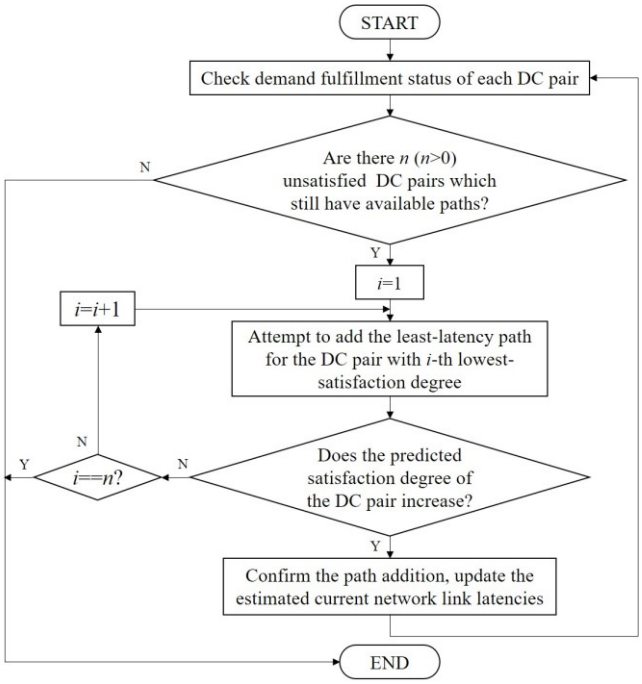

Fig. 1. Flow chart for the dynamic path adjustment.

III. EVALUATION AND DISCUSSION

We evaluate DFSM in terms of the latency performance of each DC pair and the system flow entry number by comparing the results with the case of adopting equal-cost shortest paths during the path-finding, and by comparing the system flow entry number with the case of adopting label-based tunneling. In addition, we also compare the standard deviation of path latencies in the case of adopting even traffic splitting.

We use Mininet (mininet.org) to construct several inter-DC WAN topologies with virtual switches and hosts (as local DCs), as shown in Fig. 2. Topologies A to D (Fig. 2(a) to Fig. 2(d)) are the same as the practical topologies used by wECMP-d [8]. The capacity of each bidirectional link is 10 Gbps. For each topology, 6 numbered edge switches are selected to connect 6 DCs. Each DC sends 20 TCP flows to every other DC at random rate (1 to 3 Gbps). The DFSM will attempt to optimize each DC pair in each topology. In order to determine the performance limit, the latency demand set for each DC pair is high enough so as never to be satisfied.

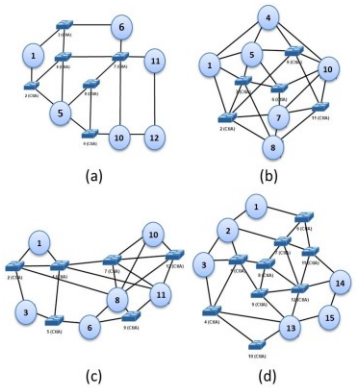

Fig. 2. Practical topologies with 30 DC pairs.

1) Compared with ECMP: DFSM reduces 10% to 48% latency by consuming 8% to 20% more flow entries. Fig. 3 shows the number of system flow entries and the corresponding average latency of all DC pairs in each topology, along with the comparison with ECMP. The results indicate that adopting k (k = 5) shortest paths (DFSM) reduces about 48%, 14% and 10% of average latency of all DC pairs results from adopting ECMP in topologies A, B and C, by consuming about 8%, 12% and 20% more flow entries, respectively. We can see that DFSM provides higher investment efficiency of flow entries in less-interconnected topologies, since there are few equal-cost shortest paths between most pairs. In general, the nodes may not be highly interconnected in a large-scale deployment as a result of the distance between the nodes and the deployment cost.

2) Compared with label-based tunneling: DFSM saves 15% to 30% flow entries. Table I shows the number of system flow entries along with the comparison with label-based tunneling; the result indicates that DFSM saves about 30% system flow entries in topologies A and D, and about 15% system flow entries in topologies B and C.

TABLE I. THE NUMBER OF SYSTEM FLOW ENTRIES

		DFSM	***label-based tunneling***
Topology	A	137	191
	B	103	121
	C	123	147
	D	142	193

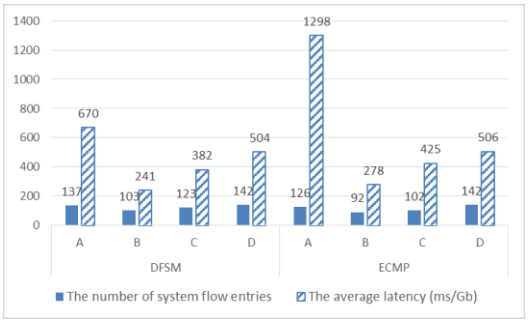

Fig. 3. DFSM vs ECMP.

3) Compared with even traffic splitting: DFSM reduces the standard deviation of path latencies from 14% to 7%. The average standard deviation of path latencies of all DC pairs is about 7% and 14% respectively of average path latency for all topologies when the latency-aware traffic splitting and the even traffic splitting is adopted.

IV. CONCLUSION

In this paper, we propose an SDN-based Dynamic Flowentry-Saving Multipath (DFSM) mechanism for inter-DC WAN traffic forwarding. Our evaluations indicate that DFSM saves 15% to 30% system flow entries in different topologies compared to label-based tunneling, and also reduces average latency by 10% to 48% by consuming 8% to 20% more flow entries than ECMP in less-interconnected topologies. In addition, compared to even traffic splitting, DFSM reduces the standard deviation of path latencies from 14% to 7%.

REFERENCES

[1] D. Thaler and C. Hopps. Multipath issues in unicast and multicast next-hop selection. IETF RFC 2991, Nov. 2000.

[2] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, ‘‘A roadmap for traffic engineering in SDN-OpenFlow networks,’’ Comput. Netw., vol. 71, pp. 1–30, Oct. 2014.

[3] OpenFlow, https://www.opennetworking.org/sdn-resources/openflow.

[4] Cord. https://wiki.opencord.org/.

[5] S. Fang, Y. Yu, C.H. Foh, K.M.M. Aung, A loss-free multipathing solution for data center network using software-defined networking approach, in: APMRC, 2012 Digest, IEEE, 2012, pp. 1–8.

[6] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined wan. In SIGCOMM, 2013.

[7] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven WAN. In Proc. ACM SIGCOMM, 2013.

[8] J. Zhang, K. Xi, L. Zhang, and H. Chao, “Optimizing network performance using weighted multipath routing,” in Computer Communications and Networks (ICCCN), 2012 21st International Conference on, pp. 1–7, 30 2012-aug. 2 2012.