

A distributed load balancing architecture based on in-band network telemetry

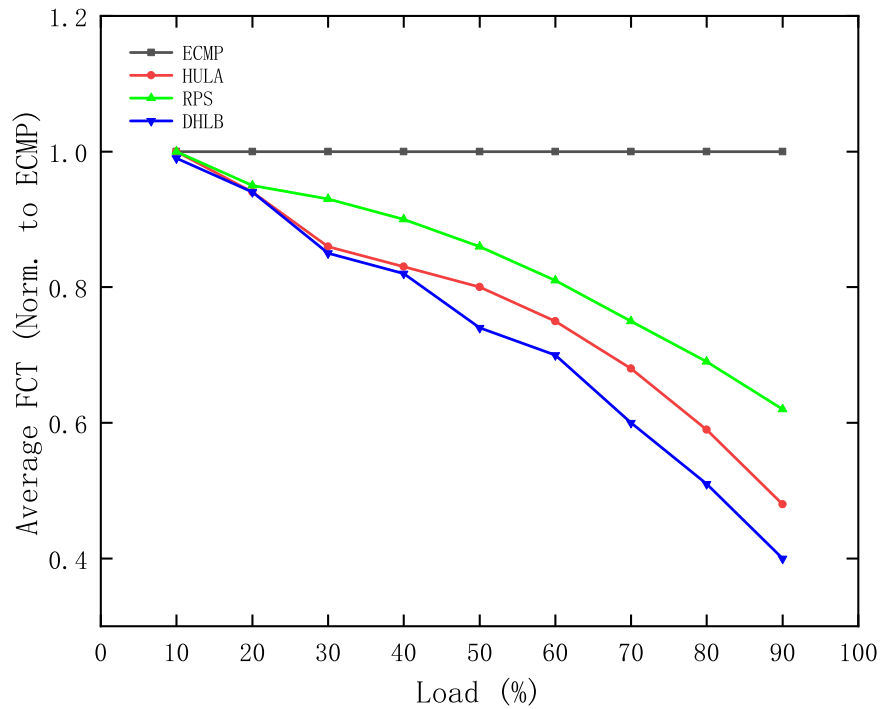
Mingfa Li¹, Huiling Shi¹, Lizhuang Tan¹, and Wei Zhang¹

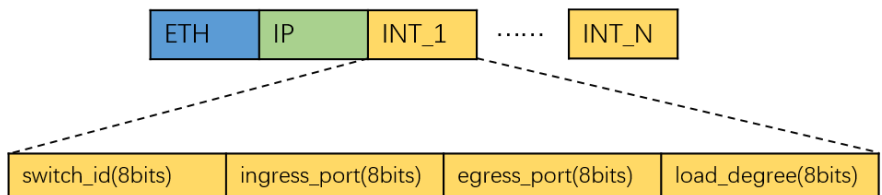
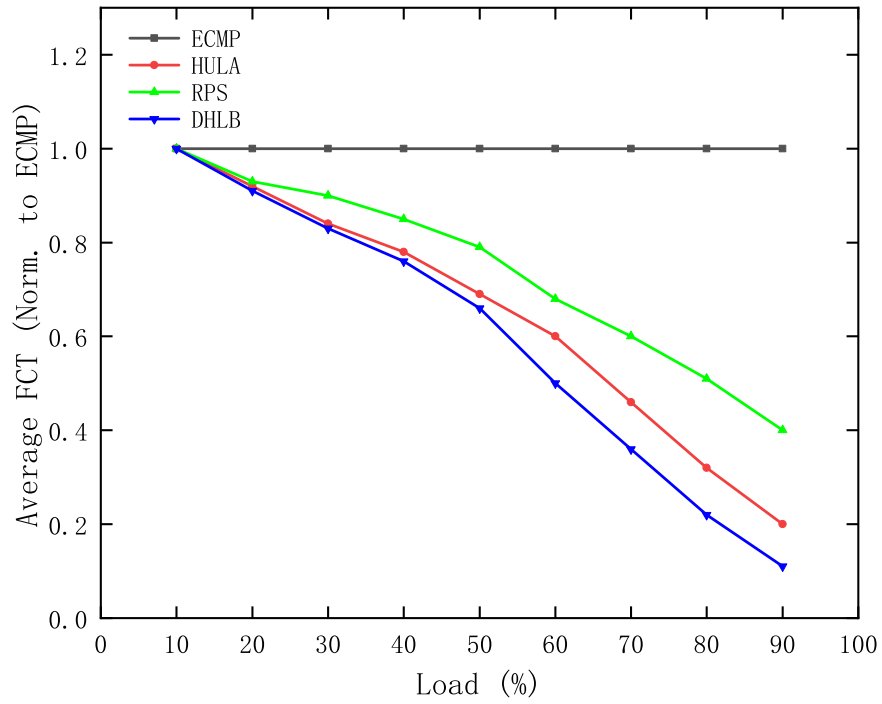
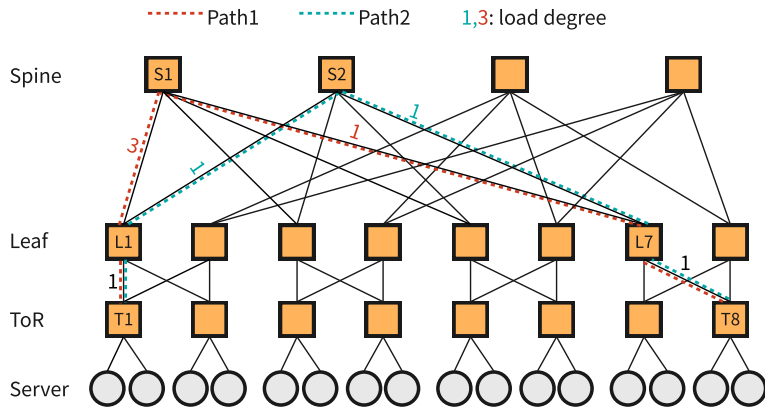
¹Shandong Academy of Sciences National Supercomputer Center In Jinan

July 16, 2024

Abstract

A data center (DC) is supposed to efficiently distribute the bandwidth of the network to provide high-quality traffic transmission. However, the load imbalance issue can easily occur due to the complex topology and traffic features. Equal-Cost Multi-Path (ECMP) distributes traffic on different paths but doesn't consider network congestion. Although HULA solved some of ECMP's problems, it can easily congest the best path. RPS randomly distributes packets on paths, which can easily lead to packet disorder in some scenarios. This paper presents DHLB, a distributed hop-by-hop load balancing architecture based on in-band network telemetry. With active In-band network telemetry, DHLB collects the necessary load information and stores it in the load information table. DHLB distributes traffic proportionally on different paths based on their load degree. We build a fat tree topology on mininet to verify the performance of our design. From experimental results, DHLB performs better than other schemes in terms of average flow complete time (FCT). It also performs better on additional overhead than another probe-based scheme.





A distributed load balancing architecture based on in-band network telemetry

Mingfa Li^{1,2} | Huiling Shi^{1,2} | Lizhuang Tan^{1,2} | Wei Zhang^{1,2}

¹ Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

² Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China

Correspondence

Huiling Shi, Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 250000, China.
Email: shihl@sdas.org

Lizhuang Tan, Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 250000, China.
Email: tanlzh@sdas.org

Abstract

A data center (DC) is supposed to efficiently distribute the bandwidth of the network to provide high-quality traffic transmission. However, the load imbalance issue can easily occur due to the complex topology and traffic features. Equal-Cost Multi-Path (ECMP) distributes traffic on different paths but does not consider network congestion. Although HULA solved some of ECMP's problems, it can easily congest the best path. RPS randomly distributes packets on paths, which can easily lead to packet disorder in some scenarios. This paper presents DHLB, a distributed hop-by-hop load balancing architecture based on in-band network telemetry. With active in-band network telemetry, DHLB collects the necessary load information and stores it in the load information table. DHLB distributes traffic proportionally on different paths based on their load degree. We build a fat tree topology on mininet to verify the performance of our design. From experimental results, DHLB performs better than other schemes in terms of average flow complete time (FCT). It also performs better on additional overhead than another probe-based scheme.

KEYWORDS

load balancing, data center network, network congestion, in-band network telemetry

1 | INTRODUCTION

As the core infrastructure of cloud computing, the data center (DC) is supposed to efficiently utilize the bandwidth of the network to provide huge throughput and high-quality traffic transmission. Multi-rooted Fat-tree and Clos are the main topologies to construct a data center and the traffic within the data center has the characteristics of high dynamic and strong burst. Due to the complex topology and traffic features, load imbalance can easily occur, leading to network congestion. To optimize this issue, a large number of load balancing methods have been proposed in the past few decades.

Equal-Cost Multi-Path (ECMP) is the typical load balancing scheme in data centers, which randomly distributes traffic on different feasible paths after performing a hash calculation on the five tuples in the packet header. ECMP is widely used due to its ease of deployment. However, few large flows account for more than 80% of realistic data center networks¹. Due to the characteristics of traffic and ECMP takes no account of congestion information and suffers from hash collisions, it might make congested paths even more congested. RPS² distributes packets on all feasible paths randomly, which improves link utilization. However, as a packet level and congestion-agnostic load balancing scheme, it may lead to packet disorder which can easily cause congestion windows to drop for TCP can't distinguish between disordered and lost packets. HULA³ only maintains the best next-hop path to the destination switch through neighboring switches. Although HULA performs better on average flow

TABLE 1 Link utilization to load degree map

Utilization	0-25%	25-50%	50-70%	70-90%	Others
Load degree	5	3	2	1	0

TABLE 2 Load information table

Destination	Path	Load Degree
ToR2	T1_3L1_2T2	4
ToR2	T1_4L2_2T2	4
...
ToR8	T1_3L1_3S1_4L7_2T8	5
ToR8	T1_3L1_4S2_4L7_2T8	5
ToR8	T1_4L2_3S3_4L8_2T8	1
ToR8	T1_4L2_4S4_4L8_2T8	1

period, we use the EWMA (Exponentially Weighted Moving Average) method, as shown in (1). Δt is the time interval from the previous packet, δ is a constant related to RTT.

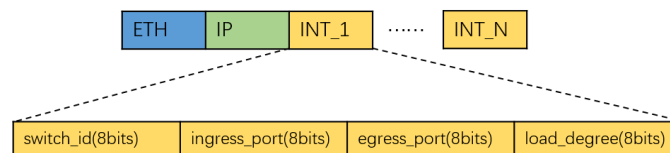
$$\mu = \delta \times (packet_{size} + \mu_{n-1}) - \Delta t \times \mu_{n-1} \quad (1)$$

To facilitate the allocation of traffic to different paths, we map link utilization to load degree based on Table I. The larger the number, the more traffic will be forwarded on this path. DHLB distributes most traffic to paths that have link utilization of less than 25% and none to paths that are greater than 90%, which can effectively avoid path congestion and fully utilize every path. The reason we handle it this way is we think we should move as much traffic on the least paths as we can and less on the congested paths. We will elaborate on the specific process afterward. A load information table stores paths to all other ToR switches, and the corresponding load degree, which is the primary basis for the routing scheme, as shown in Table II.

2.2 | Network-wide telemetry

In the beginning, DHLB obtains all the feasible paths and collects load information for switches by broadcasting an INT probe packet periodically. The probe packet is simplified and carries only necessary link and load information so that occupies little bandwidth. As shown in Figure 2, it consists of a basic IP and Ethernet header, and INT metadata field. The collected information includes the switch ID, ingress port, egress port, and load degree. In this process, all ToR switches send a probe packet to its upstream leaf switches, when the probe reaches a switch, it will update the load information table and compare the corresponding load degree with the one in the probe. Note that DHLB only reserves the smallest load degree (in other words: the largest link utilization). The specific forwarding rules are described as follows:

- For ToR switches, they are either the end or the starting point of the probe. As the starting point, they will generate the probe and send it to all connected leaf switches. As the endpoint, they will not forward the probe packet.
- For leaf switches, when they receive the probe packet from a spine switch, they will forward it to downstream ToR switches. When they receive the probe from a ToR switch, they will forward it to all connected switches except the ingress port.
- For spine switches, when they receive the probe packet, they will forward it to all connected leaf switches except the ingress port.

**FIGURE 2** Probe format of DHLB

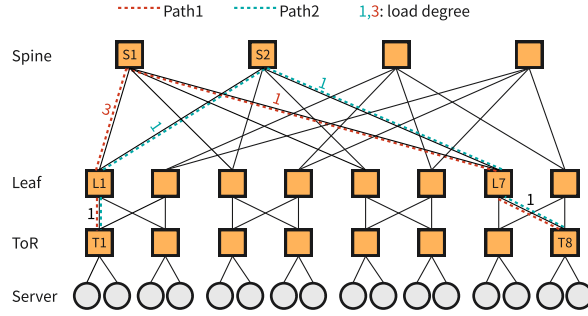


FIGURE 3 Load degree of fat-tree topology

2.3 | Routing assignment

In order to avoid packet reordering and improve link utilization, we load balance at the granularity of flowlet. A flowlet is a collection of packets with the same five tuples (src IP, dst IP, protocol type, src port, dst port) and the time gap between their reaching time is smaller than the threshold δ . For the packets in the same flowlet, they are forwarded on the same path. Otherwise, switches will make a new routing assignment for these packets. The method we use to divide different flowlets is to use a flowlet-id register and a time stamp register. The flowlet-id stores the id of each flowlet and the time stamp register stores the last timestamp for the last observed packet belonging to a flow. Once a packet arrives, the switch calculates the hash of its five tuples and queries in the flowlet-id register. If there is no record, we build a new index in the flowlet-id register and update the last arrival time. If there is a corresponding record, we subtract the current time from the last arrival time in the time stamp register and compare the result with τ . When the result exceeds τ , we build a new index in the flowlet-id register. Otherwise, we assign the same flowlet ID as the previous packet.

2.4 | Probe frequency regulating

In DCNs, when the network is under low load or the load balancing condition is stable, it's able to achieve efficient traffic transmission even if there is no load balancing scheme intervening. On the contrary, it is necessary to implement a suitable load balancing scheme. The frequency of updating the load information table depends on the sending frequency of the probe packet in this paper. Appropriately regulating the sending frequency will decrease unnecessary bandwidth consumption while ensuring efficiency. So we designed a scheme to improve this issue based on the overall load-balancing condition of the network. DHLB does this work in the control plane. Firstly, ToR Switches send their link utilization rates to the controller periodically. Then the controller will calculate the average load of the network σ . It is inversely proportional to the sending frequency F_{probe} as shown in (2), where θ is a constant.

$$F_{probe} = \frac{1}{T_{probe}} = \frac{\sigma}{\theta} \quad (2)$$

In real networks, there may be some particularly small T_{probe} that are even smaller than the flowlet threshold δ , which will lead to too fast traffic rerouting. This will not only cause low transmission reliability but also make packets disorder so that the network is filled with retransmission packets and reduces throughput. So DHLB set the minimum value of T_{probe} to twice δ .

3 | PERFORMANCE EVALUATION

In this section, we conduct experimental evaluations on the DHLB and compare the results with ECMP, HULA, and RPS. We test them in the same environment using p4 language. Our experiments will answer the following questions:

- How is the bandwidth consumed by DHLBs probes compared to other schemes using probes?

- How does DHLB perform in transmission quality compared to other schemes?

3.1 | Experimental setup

The virtual network environment is a fat-tree topology built by Mininet, consisting of four spine switches, eight leaf switches, and eight ToR switches. The interconnection way between switches is like Figure 3. To achieve customization of packets headers, we use the simple switch model of BMv2 to implement programmable switches. The bandwidth between two switches is set to 200Mbps and the bandwidth between a ToR switch and a host is set to 100Mbps. To evaluate the performance in typical data centers, we use the realistic web search workload to generate simulated traffic, which comes from realistic data centers. This type of workload is intensive: most traffic is smaller than 10KB, while a few large traffic accounts for a significant portion of the traffic. According to⁶, too large δ will result in overly coarse granularity while small δ makes frequent routing assignments. Thus we set the first reaching time gap threshold δ as the RTT of the network.

3.2 | Probe overhead

We compared the additional bandwidth consumed by probes of HULA and DHLB. HULA set its frequency as $200\mu s$. According to⁷, the overhead of probe O is shown in eq 2, where λ is the probe number a ToR will receive in a period, and numToRs is the total number of ToR switches.

$$O = \frac{probeSize \times numToRs \times \lambda}{probeFreq \times linkBandwidth} \quad (3)$$

As shown in Figure 4, HULA maintains a fixed overhead since it doesn't adjust according to load conditions. DHLB has smaller probe packets and lower sending frequency, therefore, its probe overhead remains at a low level under low loads. Even at 90% load, the additional bandwidth consumed by the probe is only similar to that of HULA. We can conclude that DHLB outperforms at probe overhead in real networks, that's why we choose dynamic frequency.

3.3 | FCT performance

The average flow completion time reflects the throughput of the network. We compared DHLB with ECMP, HULA, and RPS using FCT as our primary performance metric. Figure 5(a)-(c) shows the result as the network load changes in different flow scales. We have normalized other schemes to ECMP. When the flows are small and the load is low, the performance difference between these load balancing methods is not significant because there is sufficient available bandwidth to tolerate congestion-oblivious schemes. As the load increases, ECMP performs worse in small flows than other schemes and intolerably poorly in large flows because loads balance at flow granularity. It also suffers from congestion and hash collision. The other three schemes perform almost the same under low load, however, when the load is high, RPS performs worse than HULA and DHLB, because random spraying at packet granularity causes many disorder packets. As probe-based schemes, although DHLB and

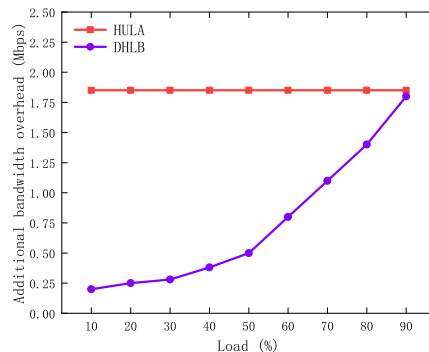


FIGURE 4 Probe overhead comparison

HULA both get worse in large and small flows with the load increases, DHLB still performs about 7% better than HULA, because DHLB avoids best path congestion by distributing traffic proportionally on different paths rather than the best.

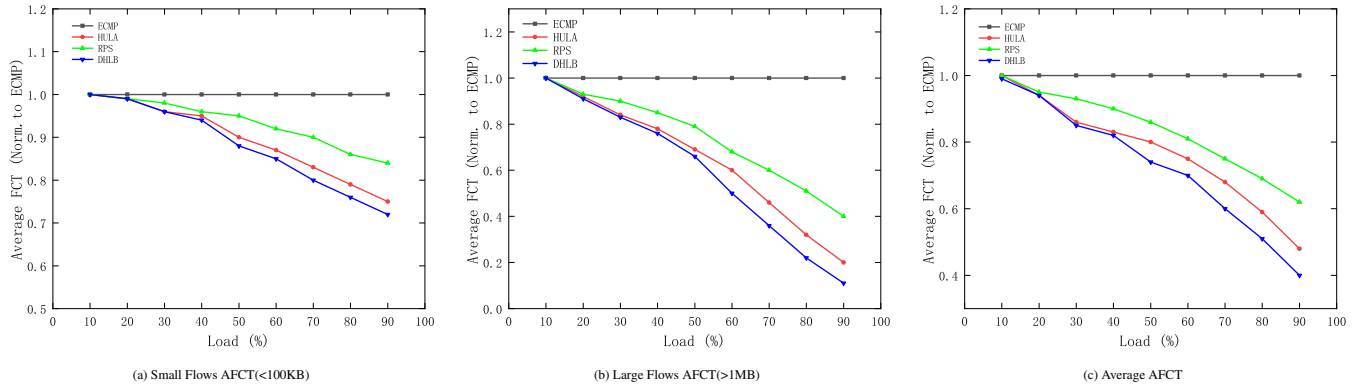


FIGURE 5 Average FCT performance under varied workload

4 | CONCLUSIONS

In this paper, We propose DHLB(a distributed hop-by-hop load balancing archetucture based on in-band network telemetry), an efficient load balancing scheme that distributes traffic proportionally on different paths based on congestion degree. DHLB periodically broadcasts probes to obtain network-wide congestion information and make routing assignments on every switch. The experiment result shows while DHLB performs effectively in load balancing against other famous schemes, it also decreases additional overhead by adjusting the sending frequency of the probe.

ACKNOWLEDGMENTS

This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant No.ZR2022LZH015, No.ZR2022QF070, and No.ZR2021LZH001, the Pilot Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2022JBZ01-01, the Taishan Scholar Program of Shandong Province in China under Grant No.TSQN202306258, No.TSQN202312230, the Project of Key R&D Program of Shandong Province under Grant No.2022CXGC020106, and the Jinan Scientific Research Leader Studio Project under Grant No.2021GXRC091.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Baburao D, Pavankumar T, Prabhu C. Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. *Applied Nanoscience*. 2023;13(2):1045–1054.
2. Dixit A, Prakash P, Hu YC, Kompella RR. On the impact of packet spraying in data center networks. In: IEEE. 2013:2130–2138.
3. Katta N, Hira M, Kim C, Sivaraman A, Rexford J. Hula: Scalable load balancing using programmable data planes. In: ACM. 2016:1–12.
4. Tan L, Su W, Zhang W, et al. In-band network telemetry: A survey. *Computer Networks*. 2021;186:107763.
5. Tan L, Su W, Miao J, Zhang W. FindINT: Detect and locate the lost in-band network telemetry packet. *IEEE Networking Letters*. 2021;4(1):20–24.
6. Javadpour A, Sangaiah AK, Pinto P, et al. An energy-optimized embedded load balancing using DVFS computing in cloud data centers. *Computer Communications*. 2023;197:255–266.