


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}

¹Key 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

²Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan, China

Correspondence

Huiling Shi and 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: shihl@sdas.org and tanlzh@sdas.org

Funding information

Shandong Provincial Natural Science Foundation, Grant/Award Numbers: 2023QF025, ZR2022LZH015, ZR2022QF070, ZR2021LZH001; Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences), Grant/Award Number: 2023PX057; Talent Project of Qilu University of Technology (Shandong Academy of Sciences), Grant/Award Number: 2023RCKY141; Shandong Provincial Soft Science Key Project in the field of Cybersecurity and Informatization; Pilot Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences), Grant/Award Number: 2022JBZ01-01; Taishan Scholar Program of Shandong Province in China, Grant/Award Numbers: TSQN202306258, TSQN202312230; Project of Key R&D Program of Shandong Province, Grant/Award Number: 2022CXGC020106; Jinan Scientific Research Leader Studio Project, Grant/Award Number: 2021GXRC091

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 across multiple paths, potentially causing packet reordering in certain 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 gathers essential load information and systematically records it in a 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. Experimental results indicate that DHLB outperforms other schemes regarding average Flow Completion Time (FCT). It also performs better on additional overhead than another probe-based scheme.

KEYWORDS

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

1 | INTRODUCTION

As the backbone of cloud computing, data centers (DCs) must efficiently utilize network bandwidth. 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.¹ Due to the topology and traffic features, load imbalance can easily occur. 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 a standard load balancing scheme in data centers that distributes traffic across multiple feasible paths by hashing the five-tuple in the packet header. ECMP's widespread adoption is largely due to its ease of deployment. However, a small number of large flows dominate, accounting for over 80% of traffic in real-world data center networks.² Due to traffic characteristics and the fact that ECMP does not consider congestion information and is prone to hash collisions, it may exacerbate congestion on already congested paths. RPS³ randomly distributes packets across all feasible paths, enhancing link utilization. However, as a packet-level, congestion-agnostic load balancing scheme, it may induce packet disorder, potentially causing TCP congestion windows to shrink since TCP cannot differentiate between disordered and lost packets. HULA⁴ maintains only the best next-hop path to the destination switch via neighboring switches. While HULA outperforms other schemes in average flow completion time (FCT), during sudden traffic spikes, selecting the best path may result in severe congestion.

Based on the development of Programming Protocol-independent Packet Processors (p4), In-band network telemetry (INT) was proposed in 2015. It is an emerging network measurement framework without the control plane intervening,⁵ which makes it timely. INT consists of two types: passive INT, which transmits messages through service traffic, and active INT, sending probe packets to collect customized information such as queue depth, queue delay, and link utilization.⁶ Although INT is a powerful tool for optimizing network issues, it has yet to see widespread adoption in load balancing. Leveraging INT, we can conveniently gather critical link load information in real-time.

To overcome the best path congestion and packet disorder issue, this paper proposes DHLB (a distributed hop-by-hop load balancing architecture based on in-band network telemetry), which collects load information and stores it in switches. DHLB executes routing decisions at the granularity of the flowlet based on this load information table. In summary, our main contributions include:

- We propose a distributed hop-by-hop load balancing architecture, which collects network-wide congestion information by active INT. DHLB distributes traffic on different paths proportionally based on the corresponding load degree.
- We propose a simple mechanism to dynamically regulate the sending frequency of sending probes, reducing overhead caused by probes.
- We conduct experiments on mininet, the results prove DHLB performs better than other schemes in terms of transmission quality.

2 | SYSTEM MODEL

As shown in Figure 1, DHLB consists of four parts: load information table, network-wide telemetry, routing assignment, and regulating the sending frequency. Based on the collected load metrics, switches precisely allocate traffic to maximize bandwidth utilization. We will elaborate on our design in this section.

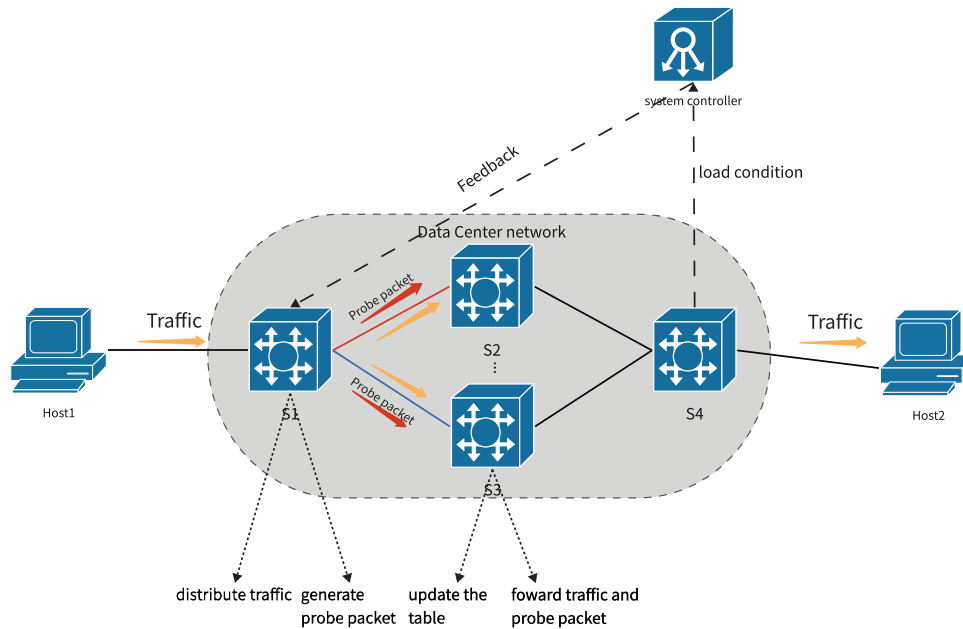


FIGURE 1 System model.

2.1 | Load information table

We design the load degree to load traffic on all available paths rather than the one best path. DHLB first calculates the ports' bandwidth utilization. Considering that the BMv_2 switch cannot directly obtain ports' bandwidth utilization rate, and can only estimate the bandwidth utilization rate by counting the size of data packets and the number of transmitted bytes within a certain 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 optimize traffic allocation across multiple paths, we map link utilization to load degree as shown in Table 1. The higher the load degree, the more traffic is assigned to that path. DHLB directs the majority of traffic to paths with less than 25% utilization and avoids paths exceeding 90% utilization, effectively preventing congestion and maximizing path usage. To ensure efficiency, we prioritize shifting traffic onto less congested paths, which have greater load-bearing capacity. The detailed process is discussed later. A load information table, shown in Table 2, records the paths to all other ToR switches and their corresponding load degrees, forming the foundation of our routing scheme. The role of the load information table in the routing assignment phase is mainly the reference for routing decisions.

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, each ToR switch sends a probe packet to its upstream leaf switches. Upon reaching a switch, the probe updates the load information table and compares the corresponding load degree with the value in the probe. DHLB retains only the smallest load degree (in other words: the highest link utilization). The specific forwarding rules are outlined 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.

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 |

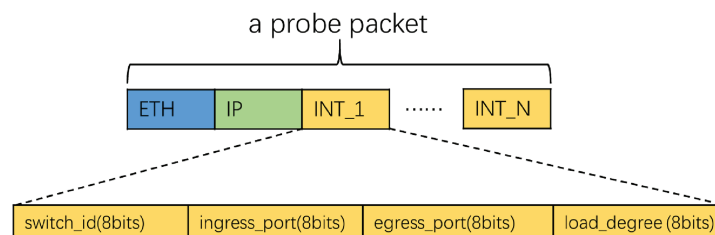


FIGURE 2 Probe format of DHLB.

- 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.

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 operates within the control plane. Initially, ToR switches periodically report their link utilization rates to the controller, which subsequently calculates the average network load 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 DHLB's 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. According to⁷ 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.

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

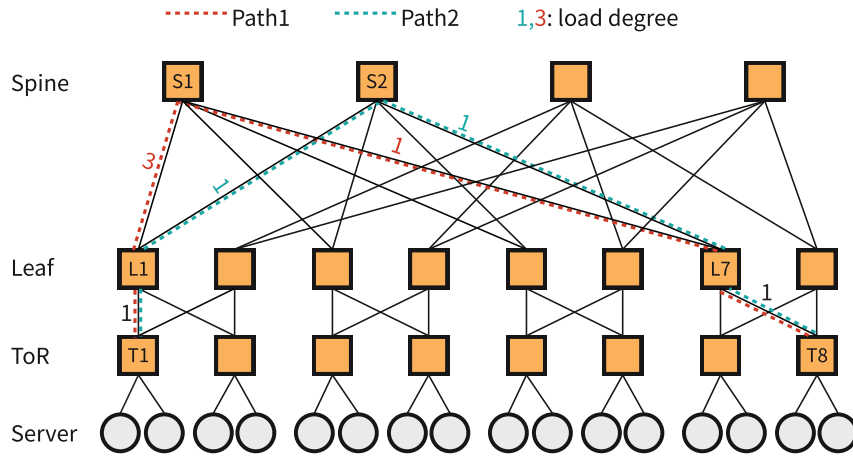


FIGURE 3 Load degree of fat-tree topology.

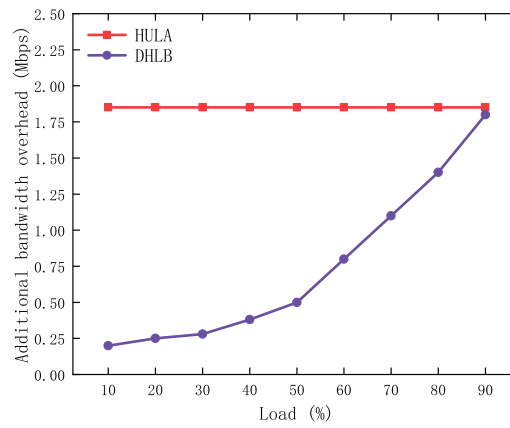


FIGURE 4 Probe overhead comparison.

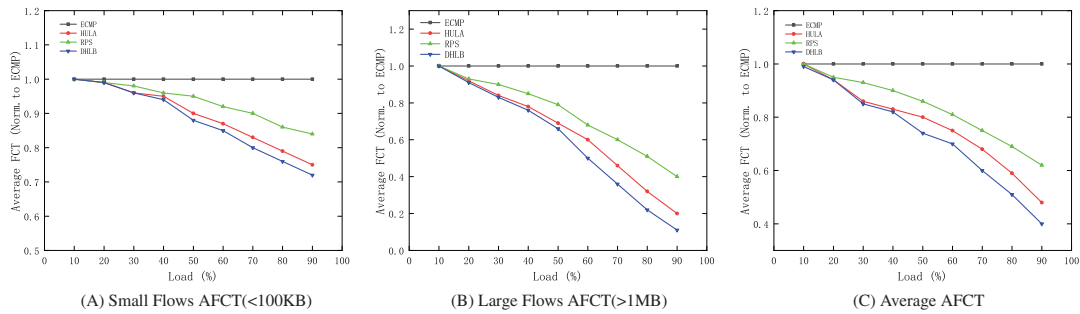


FIGURE 5 Average FCT performance under varied workload.

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.

The average Flow Completion Time (FCT) serves as an indicator of network throughput. We evaluated DHLB against ECMP, HULA, and RPS, using FCT as the primary performance metric. Figure 5A–C presents the results as network load varies across different flow scales.

3.3 | FCT performance

The average Flow Completion Time (FCT) serves as an indicator of network throughput. We evaluated DHLB against ECMP, HULA, and RPS, using FCT as the primary performance metric. Figure 5A–C presents the results as network load varies across 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 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.

4 | CONCLUSIONS

In this paper, We propose DHLB(a distributed hop-by-hop load balancing architecture 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 Nos. 2023QF025, ZR2022LZH015, ZR2022QF070, and ZR2021LZH001, the Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant No. 2023PX057, the Talent Project of Qilu University of Technology (Shandong Academy of Sciences) under Grant No. 2023RCKY141, Shandong Provincial Soft Science Key Project in the field of Cybersecurity and Informatization, 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 Nos. TSQN202306258, 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.

PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/itl2.587>.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Mingfa Li  <https://orcid.org/0009-0002-6221-9334>

REFERENCES

1. Han J, Xue K, Wang W, Li R, Sun Q, Lu J. RateMP: Optimizing Bandwidth Utilization With High Burst Tolerance in Data Center Networks. *INFOCOM'24*. IEEE; 2024: 1361–1370.
2. Baburao D, Pavankumar T, Prabhu C. Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. *Appl Nanosci*. 2023;13(2):1045–1054.
3. Dixit A, Prakash P, Hu YC, Kompella RR. On the Impact of Packet Spraying in Data Center Networks. *INFOCOM'13*. IEEE; 2013: 2130–2138.
4. Katta N, Hira M, Kim C, Sivaraman A, Rexford J. Hula: Scalable Load Balancing Using Programmable Data Planes. *SOSR'16*. ACM; 2016:1–12.
5. Tan L, Su W, Zhang W, et al. In-band network telemetry: a survey. *Comput Netw*. 2021;186:107763.
6. Tan L, Su W, Miao J, Zhang W. FindINT: detect and locate the lost in-band network telemetry packet. *IEEE Networking Lett*. 2021;4(1):20–24.
7. Javadpour A, Sangaiah AK, Pinto P, et al. An energy-optimized embedded load balancing using DVFS computing in cloud data centers. *Comput Commun*. 2023;197:255–266.

How to cite this article: Li M, Shi H, Tan L, Zhang W. A distributed load balancing architecture based on in-band network telemetry. *Internet Technology Letters*. 2024;e587. doi: 10.1002/itl2.587