ZooRoute: Enhancing Cloud-Scale Network Reliability via Overlay Proactive Rerouting

Xiaoqing Sun[1†], Xionglie Wei[1†], Xing Li[1,2†], Ju Zhang[1], Bowen Yang[1], Yi Wang[1], Ye Yang[1], Yu Qi[1], Le Yu[1], Chenhao Jia[1], Zhanlong Zhang[1], Xinyu Chen[1], Jianyuan Lu[1], Shize Zhang[1], Enge Song[1] Song Yang[1], Tian Pan[1], Rong Wen[1], Biao Lyu[1], Yang Xu[3], Shunmin Zhu[1,4]
[1]Alibaba Cloud [2]Zhejiang University [3]Fudan University [4]Hangzhou Feitian Cloud
alibaba_cloud_network@alibaba-inc.com

Abstract

This paper presents *ZooRoute*, a tenant-transparent, fast failure recovery service that requires no modifications to physical devices. ZooRoute leverages the overlay layer and enables traffic flows to bypass failures by altering source ports (*srcPorts*) in packet headers during encapsulation. To enable deployment in large-scale cloud networks, ZooRoute proposes: 1) On-demand probing to efficiently monitor a vast number of hosts while minimizing telemetry costs. 2) Table compression to record the states of numerous paths with limited on-chip resources. 3) A device-sensing mechanism to prevent unnecessary reconnections in stateful forwarding. Deployed in Alibaba Cloud for 18 months, ZooRoute has significantly improved network reliability, reducing cumulative outage time by 92.71%.

CCS Concepts

• **Networks** → **Cloud computing**; **Network reliability**; **Network management**.

Keywords

Overlay, Proactive Rerouting, Network Reliability, ECMP

ACM Reference Format:
Xiaoqing Sun, Xionglie Wei, Xing Li, Ju Zhang, Bowen Yang, Yi Wang, Ye Yang, Yu Qi, Le Yu, Chenhao Jia, Zhanlong Zhang, Xinyu Chen, Jianyuan Lu, Shize Zhang, Enge Song, Song Yang, Tian Pan, Rong Wen, Biao Lyu, Yang Xu, Shunmin Zhu. 2025. ZooRoute: Enhancing Cloud-Scale Network Reliability via Overlay Proactive Rerouting. In *ACM SIGCOMM 2025 Conference (SIGCOMM '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3718958.3750483

1 Introduction

As more applications move to the cloud, network reliability has become a key concern for Cloud Service Providers (CSPs) and their tenants. Failures are inevitable in large scale networks [1]. However, existing recovery solutions struggle to achieve timely global bypass or impose significant burdens on tenants. Specifically, when failures occur, strategies like fast reroute [2] and traffic engineering [3] can only offer local bypass in seconds or global bypass in minutes. Though protective rerouting [4] and network architecture simplifications [5–8] are proposed to accelerate global reconvergence, they require upgrades to underlying equipment, making large-scale deployment challenging. As a result, tenants are forced to develop their own recovery solutions, which typically involve redundant resources or protocol stack modifications, thereby increasing capital and operating expenses.

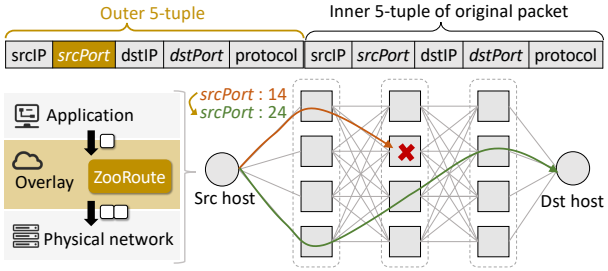

Figure 1: Key insight of ZooRoute

This paper introduces ZooRoute, a fast failure recovery service that ensures global bypass in large-scale cloud networks within seconds. As illustrated in Fig. 1, ZooRoute runs on the host side, and leverages the overlay layer to induce path alternation in physical networks during packet encapsulation. Specifically, with the standard Equal Cost Multi-Path (ECMP) routing on physical switches, different values of the "source port" (*srcPort*) in the "outer" header may yield different hash results, directing packets to different network paths. Therefore, ZooRoute monitor network path status by sending probes with varied outer *srcPorts*. When failures occur, it instantly reroute traffic to a working path by changing the failed *srcPort* (*e.g.,* 14) to an available one (*e.g.,* 24). This "proactive telemetry & path altering" mechanism is tenant-transparent and requires no modifications to physical devices. Besides, several strategies are developed to facilitate ZooRoute in large-scale CSP networks.

2 ZooRoute Design

We design ZooRoute as a host-side distributed system to pursue faster response and avoid single-point of failure. With a software-hardware architecture, its workflow comprises: *i) Path probing*: the prober periodically sends request packets with different *srcPorts* to other hosts. *ii) Path recording*: the analyzer deal with the response packets to record information like *dstIP*, *srcPort*, and status. *iii) Path altering*: when encapsulating packets, the hypervisor selects available *srcPorts* in the path table to bypass failures.

[†]Co-first authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCOMM '25, Coimbra, Portugal
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1524-2/25/09
https://doi.org/10.1145/3718958.3750483

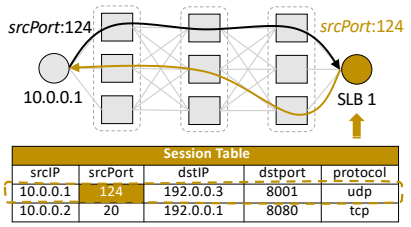

Figure 2: *SrcPort* learning in stateful forwarding scenarios like SLB.

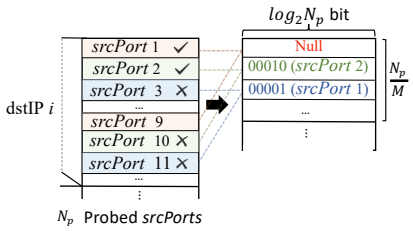

Figure 3: Path table compression with hierarchical indexing

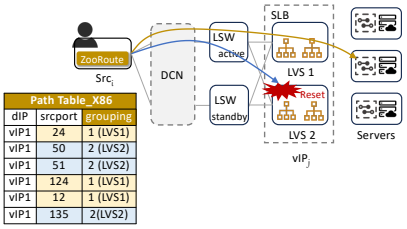

Figure 4: Path altering in stateful forwarding w/wo device grouping

Path probing. Comprehensive telemetry is crucial for timely network status monitoring, but in large networks, full-mesh and high-frequency probing consume lots of resources. ZooRoute employs on-demand probing, which includes: *i) Active IP probing*: ZooRoute initiates a telemetry task on $host_i$ to $host_j$ only if there is tenant traffic in between. *ii) Active srcPort probing*: ZooRoute employs full probing mode and partial probing mode. While the former conducts periodic probings of all *srcPorts*, the latter only probes those currently used by tenant traffic. ZooRoute runs in partial probing under normal conditions and shifts to full probing when network failures are detected. *iii) SrcPort learning in stateful forwarding*: ZooRoute conducts round-trip telemetry, where an available *srcPort* means both the forward and reverse paths are functional. As shown in Fig. 2, stateful network elements can leverage session information to passively find working paths based on the *srcPorts* used by their counterparts, thus halving CPU and bandwidth consumptions.

Path recording. Efficient path recording is essential for rerouting traffic to functional paths at oneshot. However, large CSPs apply hardware to accelerate packet processing, whose on-chip memories are always limited [9–12]. As shown in Fig. 3, for the probed N_p *srcPorts* of $dstIP_i$, ZooRoute groups every M *srcPorts* to computes a hash value, using as a entry index to the path table.

- If all *srcPorts* in this group work well, the corresponding table entry will remain empty (depicted in red)
- If "bad" *srcPorts* exist, we fill the entry with a randomly selected working *srcPort* in this group (depicted in green)
- If all *srcPorts* are unavailable, a working one from other groups will be randomly chosen (depicted in blue)

Consequently, when a packet arrives, it also computes the index by hashing a group of M *srcPorts*. If the corresponding entry is empty, it can randomly utilize any of these M *srcPorts*. Otherwise, only the recorded one can be used.

Path altering. While bypassing failures, path alteration may disrupt packet transmission. In stateful forwarding, path changes can lead the same flow to different devices, causing re-connections due to the lack of session information. To address this, ZooRoute arms stateful forwarding with a device sensing mechanism by adding a "Backend ID" field in telemetry packets. This field is filled by backend hosts with their MAC address as a unique identifier, allowing the source host to group *srcPorts* when receiving telemetry responses (*e.g.,* *srcPorts* 24/124/12 and 50/51/135 in Fig. 4). When failure occurs, ZooRoute prioritizes replacing the current $srcPort_j$ with those from the same group, so as to ensure the rerouted flow still reaches the same backend host.

3 Preliminary Results

We evaluate the effectiveness of ZooRoute's optimization strategies by collecting data from three regions in AliCloud with $100k+$, $50k+$ and $10k+$ hosts, respectively. Then, we present ZooRoute's online overhead and performance.

Evaluation of optimizations. For path probing, we regard the number of telemetry packets as costs. Compared to full-mesh probing, active IP probing saves 60%~70% of telemetry costs, and active *srcPort* probing further reduces the overall cost to around 5%. Instead of has great impacts on regional total costs, the *srcPort* learning mechanism lowers the host-granularity maximum number of telemetry packets. For path recording, compared to storing an available *srcPort* in every entry of a P4 gateway's built-in hash table, hierarchical indexing achieves ~25x compression. For path altering, We deploy an SLB instance with backend HTTP service in AliCloud, and simulate ZooRoute's reactions during failures by letting the hypervisor assign a different *srcPort* to every three packets of the same flow. Without device sensing, about 25% of requests exhaust their maximum retries and ultimately fail, while with device sensing, all requests are successfully served in oneshot.

Online overhead. We assess overhead of ZooRoute by measuring the P99 resource consumption of hosts in the largest region. Results show that, for a server, ZooRoute uses 2% ~3% CPU, 3% memory and less than 8.5×10^{-4} of bandwidth. For a P4-based cloud gateway, ZooRoute occupies less than 7.5% CPU, 3% SRAM and 9.5×10^{-6} bandwidth. Compared to reputation and revenue losses caused by SLA violations, these overheads are acceptable for large CSPs.

Online performance. ZooRoute has been deployed in AliCloud for 18 months. By comparing the packet loss rates of probings and tenant traffic, we find it has reduced the outage time by 92.71%. This reduction is consistent, with monthly ratios over 75% and regional ratios over 80%. Variations are caused by the nature of different outages. The Top3 failure types with largest cumulative reductions are listed in Tab. 1. ZooRoute has masked 98% of failures from tenants noticing. We investigate cases where it failed to achieve rapid recovery, and find that the main reasons are severe network capacity loss and the unstable status of faulty physical devices.

Table 1: Top3 failure types with largest cumulative reductions by ZooRoute

Root causes	Avg. red. time (s)	Fraction (%)
Jitter in long-haul link	7.47	84.67
Switch card or port exception	139.52	93.12
Fiber issue or cutover in ISP	295.33	98.60

References

[1] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. Understanding network failures in data centers: measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 Conference*, pages 350–361, 2011.

[2] Ping Pan, George Swallow, and Alia Atlas. Fast reroute extensions to rsvp-te for lsp tunnels. Technical report, 2005.

[3] Zhizhen Zhong, Manya Ghobadi, Alaa Khaddaj, Jonathan Leach, Yiting Xia, and Ying Zhang. Arrow: restoration-aware traffic engineering. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 560–579, 2021.

[4] David Wetherall, Abdul Kabbani, Van Jacobson, Jim Winget, Yuchung Cheng, Charles B Morrey III, Uma Moravapalle, Phillipa Gill, Steven Knight, and Amin Vahdat. Improving network availability with protective reroute. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 684–695, 2023.

[5] Umesh Krishnaswamy, Rachee Singh, Nikolaj Bjørner, and Himanshu Raj. Decentralized cloud wide-area network traffic engineering with {BLASTSHIELD}. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 325–338, 2022.

[6] Marek Denis, Yuanjun Yao, Ashley Hatch, Qin Zhang, Chiun Lin Lim, Shuqiang Zhang, Kyle Sugrue, Henry Kwok, Mikel Jimenez Fernandez, Petr Lapukhov, et al. Ebb: Reliable and evolvable express backbone network in meta. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 346–359, 2023.

[7] Umesh Krishnaswamy, Rachee Singh, Paul Mattes, Paul-Andre C Bissonnette, Nikolaj Bjørner, Zahira Nasrin, Sonal Kothari, Prabhakar Reddy, John Abeln, Srikanth Kandula, et al. {OneWAN} is better than two: Unifying a split {WAN} architecture. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 515–529, 2023.

[8] Alexander Krentsel, Nitika Saran, Bikash Koley, Subhasree Mandal, Ashok Narayanan, Sylvia Ratnasamy, Ali Al-Shabibi, Anees Shaikh, Rob Shakir, Ankit Singla, et al. A decentralized sdn architecture for the wan. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 938–953, 2024.

[9] Janet Tseng, Ren Wang, James Tsai, Yipeng Wang, and Tsung-Yuan Charlie Tai. Accelerating open vswitch with integrated gpu. In *Proceedings of the Workshop on Kernel-Bypass Networks*, pages 7–12, 2017.

[10] Daniel Firestone, Andrew Putnam, Sambhrama Mundkur, Derek Chiou, Alireza Dabagh, Mike Andrewartha, Hari Angepat, Vivek Bhanu, Adrian Caulfield, Eric Chung, et al. Azure accelerated networking:{SmartNICs} in the public cloud. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 51–66, 2018.

[11] Peixuan Gao, Yang Xu, and H Jonathan Chao. Ovs-cab: Efficient rule-caching for open vswitch hardware offloading. *Computer Networks*, 188:107844, 2021.

[12] Tian Pan, Nianbing Yu, Chenhao Jia, Jianwen Pi, Liang Xu, Yisong Qiao, Zhiguo Li, Kun Liu, Jie Lu, Jianyuan Lu, et al. Sailfish: Accelerating cloud-scale multi-tenant multi-service gateways with programmable switches. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 194–206, 2021.