USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Skyline: A Cloud Centric Internet Monitoring Engine

Shixian Guo, *ByteDance;* Ziqian Liu, *The University of Hong Kong;* Yangyang Bai, Yuan Chen, Kefei Liu, Qi Zhang, Songlin Liu, Yang Lv, Jianwei Hu, Gen Li, Zhenyang Zhong, Sisi Wen, Yongbin Dong, Feng Luo, Anjian Chen, Rui Han, Jiale Feng, Lingpei Meng, Siwan Chen, Hang Li, Shuai Xu, Juntao Zhong, and Chaoran Hu, *ByteDance;* Yibo Huang, *University of Michigan;* Yiming Qiu, *The University of Hong Kong*

https://www.usenix.org/conference/nsdi26/presentation/guo-shixian

This paper is included in the Proceedings of the 23rd USENIX Symposium on Networked Systems Design and Implementation.

May 4–6, 2026 • Renton, WA, USA

ISBN 978-1-939133-54-0

Open access to the Proceedings of the 23rd USENIX Symposium on Networked Systems Design and Implementation is sponsored by

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

Skyline: A Cloud-Centric Internet Monitoring Engine

Shixian Guo* Ziqian Liu§* Yangyang Bai Yuan Chen Kefei Liu† Qi Zhang Songlin Liu
Yang Lv Jianwei Hu Gen Li Zhenyang Zhong Sisi Wen Yongbin Dong Feng Luo
Anjian Chen Rui Han Jiale Feng Lingpei Meng Siwan Chen Hang Li Shuai Xu
Juntao Zhong Chaoran Hu Yibo Huang‡ Yiming Qiu§

ByteDance §*The University of Hong Kong* ‡*University of Michigan*

Abstract

Cloud providers depend on the public Internet to connect tenants and their clients, yet Internet faults are a leading cause of cloud outages: in our organization, more than 60% of network incidents happen in the Internet and account for close to 80% of user-impacting events. Effectively monitoring the Internet is challenging for cloud providers because they lack direct control and visibility into Internet internals. Our key insight is to treat coverage as a first-class goal and decompose the monitoring requirements into three *coverage dimensions—traffic direction*, *incident lifecycle*, and *tenant granularity*—then resolve each independently. We present Skyline, a cloud-centric Internet monitoring system that addresses all coverage dimensions at scale by combining purpose-built dataplane hooks with lightweight software control to minimize resource overhead, shorten reaction time, and preserve non-intrusiveness. Skyline has been deployed for more than two years. In 2025, it identified over 2,000 incidents with very high precision and recall over confirmed issues, thereby significantly improving the reliability of our cloud network.

1 Introduction

The physical separation between cloud services and clients necessitates their interconnectivity through the public Internet [1,5,6,13,29,35,45]. Unfortunately, the inherent unreliability and unpredictability of the public Internet make it a frequent source of network anomalies [8,9,11,14,18,36,40]. As a major cloud provider, we observe that over 63% of our network incidents occur on the Internet rather than within our own network infrastructure. These Internet incidents are particularly perilous because they directly disrupt the reachability between cloud tenants and their clients, leading to performance degradations, SLA violations, and revenue losses. This is evidenced by their contribution to 78% of business-impacting network outages in our production environment, along with large amount of major breakdowns across the cloud industry [16,20,26,38]. Consequently, evaluating the condition of the public Internet should be considered a paramount task for cloud providers.

Monitoring and diagnosing the public Internet is challenging for cloud providers [24,28,32,47,52], not least because they do not have direct ownership or operational control over the Internet service providers (ISPs)— switch logs as an example are simply not available. As a result, the cloud must treat the Internet as a black-box whose internal signals and knobs are not readily accessible [7,30,31,39,54]. The difficulty is further amplified by the Internet's vast scale spanning ∼200K Autonomous Systems (ASes), heterogeneity across ISPs and devices, and dynamism from constant topology changes and routing policy updates [4,19,27,42,50], which together produce a wide variety of incident types [28,32,47,52]. Consequently, datacenter monitoring methods designed for centrally controlled fabrics do not generalize to the decentralized, interdomain Internet environment, prompting cloud providers to develop specialized Internet monitoring engines.

We formalize our question as follows: *How can a cloud provider achieve high coverage of Internet incidents with limited visibility into Internet internals?* Our approach is to treat coverage as a first-class goal and *decompose* it into dimensions that align with levers cloud providers can control. In particular, we rely on three dimensions to address almost all Internet incidents in our production environment:

- **Traffic direction coverage.** Inconsistencies across closely related paths often reveal the most elusive incidents. For example, traffic can reach one cloud subnet but not an adjacent one is a strong indictor of BGP hijacking. This necessitates *fine-grained, bidirectional* reachability checks that include both outbound probing toward client address spaces, and inbound probing toward cloud subnets. Providing bidirectional coverage at scale in turn calls for probing methods that minimize resource consumption while preserving transparency to both cloud tenants and their clients.
- **Incident lifecycle coverage.** Handling network incidents is a multi-phase process that spans *incident detection*, *path rerouting*, *repair detection*, and *path recovery*. Beyond de-

*Both authors contributed equally to the paper.
†Kefei Liu is the corresponding author.

tecting failures, cloud providers must also detect *repairs* promptly so they can safely switch traffic back from backup paths to restored primary paths. Doing so requires continuous monitoring of both the normal and faulty paths, which is non-trivial because the cloud naturally steers traffic (and thus default probes) onto the normal path. Monitoring the faulty path therefore requires explicit, flexible path control.

- **Tenant-level coverage.** Cloud providers must also discern public-Internet incidents using *per-tenant, upper-layer* (layer 4–7) signals (e.g., TCP session metrics). Such signals can reveal incidents that basic reachability checks miss, and can also clarify the Internet's role in tenant-reported problems that are otherwise hard to assess. However, collecting tenant-level metrics often requires per-packet visibility, which can incur substantial resource and processing overhead if not designed carefully. Turning these signals into actionable diagnoses further demands careful system design to control noise and minimize false positives.

These dimensions define what effective cloud-based Internet monitoring must address. Measured against them, existing practices [28, 48, 52, 53] provide only partial coverage: they typically apply outbound probing but not inbound, detect incidents but not repairs, and capture layer 3 reachability but not per-tenant metrics. In our environment, naïvely adopting such designs leaves major incident classes undetected. Extending them is likewise impractical: (i) covering cloud subnet reachability with only outbound probing requires deploying agents in each target cloud subnets, driving substantial resource overhead; (ii) repair detection without flexible path control devolves to waiting for ISP announcements, incurring delays up to hours; and (iii) per-tenant, upper-layer visibility requires collecting and analyzing business traffic at scale, which layer 3 probing systems do not support.

In this paper, we propose Skyline, a cloud-centric Internet monitoring engine with three specialized subsystems, each addressing a coverage dimension. First, to cover both client and cloud reachability, *BiProbe* implements two probing systems, one for outbound and the other for inbound. Both are non-intrusive and resource-efficient by leveraging client side CDN nodes and cloud side programmable switches. Second, to cover incident lifecycle, *FlexPath* leverages cloud side segment routing to implement a light-weight path steering mechanism, allowing the concurrent monitoring of multiple candidate paths. Third, to cover tenant-level issues, *FlowHunter* obtains per-tenant, upper-layer network metrics by fully duplicating business traffic with optical splitters, followed by aggregating and analyzing per-flow information for tenant-level incident detection and tenant report self-attestation.

We have deployed Skyline in our global Internet infrastructure for over 2 years, covering all cloud traffic going through more than 40 Points of Presence (PoPs) and their associated 200 public Internet lines. In 2025, Skyline accounted for more than 2000 incidents with real-world business impact, covering more than 99% of all ultimately confirmed Internet incidents

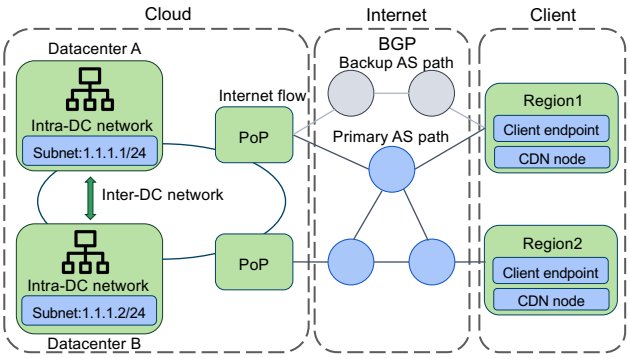


Figure 1: An end-to-end view of the cloud network.

that affected our global infrastructure. It also reached a precision rate of up to 97%, delivering a minimal amount of false positives. Since its deployment, Skyline has proven indispensable to our network diagnosis teams in ensuring the reliability of cloud network and the stability of public cloud services.

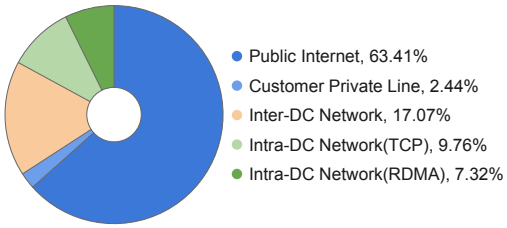
2 Motivation

In this section, we ground the Internet monitoring problem in a typical cloud network (Figure 1). Within each cloud datacenter, cloud servers communicate with each other via either TCP or RDMA-based *intra-datacenter network*. Cloud datacenters in different regions are further inter-connected together via a private backbone called *inter-datacenter network*, which connects to the public Internet at ISP entrances of fixed PoPs. The PoPs are operated by the cloud provider, offering key services such as address translation, load balancing, and firewalls. Beyond the PoPs, the public Internet consists of ASes run by ISPs. They exchange routes via the BGP protocol, and forward traffic according to router configurations, collectively forming the topology of the public Internet.

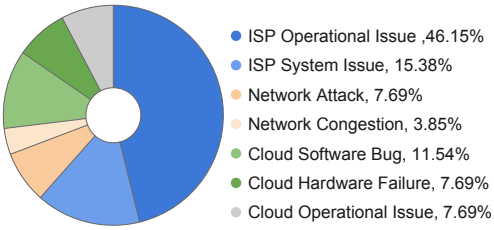
Apart from the cloud PoPs, Internet is simultaneously connected to a broad range of clients, including local endpoints and CDN servers. As a result of this network architecture, traffic between the cloud and the clients is usually way-pointed through multiple hops of public Internet ASes (i.e., AS paths). We denote public Internet as the externally managed *cloud-to-client* communication fabric, integral to service delivery but not under the cloud provider's control. From the perspective of a public cloud provider, successful cloud-to-client communication requires end-to-end health assurance, which should account for not only the cloud data center network and the clients being served, but also the public Internet in the middle. Consequently, cloud providers must take an active role in monitoring and diagnosing Internet incidents.

2.1 Taxonomy of Cloud Network Incidents

In reality, we have observed that every components of the cloud network could trigger major incidents. To understand the prevalence of different types of cloud network incidents and their implications, we present a large-scale study on all the cloud network incidents experienced by our organization,



(a) Cloud network incidents



(b) Public Internet incidents

Figure 2: A taxonomy of cloud network incidents.

a major public cloud provider, during the year of 2024. As shown in Figure 2a, The cloud intra-datacenter network in total triggered 17.08% of network incidents. More concretely, 7.32% of the cloud network incidents were caused by the RDMA-based high performance interconnects, while the rest 9.76% were linked to the TCP-based channels. The cloud inter-data center network caused 17.07% of all network incidents, almost on par with its intra-datacenter counterpart. The cloud-to-client network triggered all the remaining 65.85% of reported issues. Specifically, only 2.44% was due to anomalies in customer private lines, while all the other 63.41% were attributed to the public Internet. This shows that Internet is the dominating source of cloud network incidents, and Internet reliability is of paramount importance to the cloud.

To understand the root causes of public Internet incidents in the cloud, we present a detailed taxonomy in Figure 2b. Among all the Internet incidents, up to 46.15% were marked as ISP operational issues, meaning they were caused by ISP runtime operations such as BGP configuration updates. Another 15.38% of incidents were characterized as ISP system issues, most of which were physical link failures or router software bugs. Putting together, ISP-related issues account for 61.53% of Internet incidents. Notably, ISP-related issues could be considered *uncontrollable* from the perspective of cloud providers, because cloud cannot directly take actions to actively mitigate them. There are also other types of uncontrollable issues originated from network attacks (7.69%) and network congestion (3.85%), bringing the total to ∼73%. The rest ∼27% of issues are considered *controllable* incidents—they appear as Internet incidents, but are triggered at the PoPs controlled by the cloud provider. This includes 11.54% of cloud software bugs, 7.69% of cloud hardware failures, and 7.69% of cloud operational issues.

2.2 Observation: Coverage Dimensions

To understand why the three coverage dimensions are essential for addressing public Internet incidents, we present our

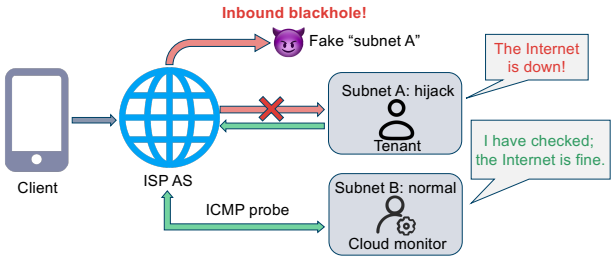


Figure 3: Example of uni-directional incidents: BGP hijacking

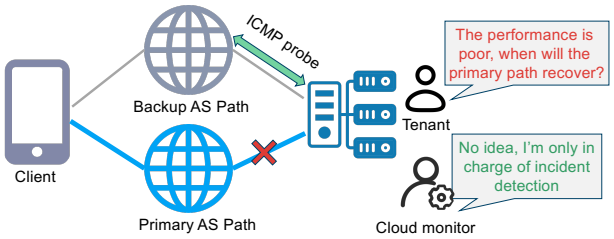


Figure 4: Primary and backup AS path comparison

observations from real world cases.

Fine-grained, bidirectional reachability. A public cloud comes with numerous subnets, each containing hundreds of public IP addresses assigned to different tenants. These subnets advertise their routes to the Internet via the BGP protocol. Similarly, client endpoints also reside in many different IP ranges, each with distinct routing preferences. Now consider a real world incidents caused by BGP route hijacking (Figure 3), where packets are forwarded to the wrong part of the Internet then getting either black-holed or controlled by an attacker. Since BGP hijacking can be uni-directional, it is possible that traffic from cloud to the clients is normal, but the opposite direction is hijacked. Moreover, since BGP routes are on the subnet granularity, it is also likely that traffic to one cloud subnet appears normal, while those to adjacent subnets are under attack. This necessitates *fine-grained, bidirectional probing over cloud subnets and client address spaces*, making it an essential coverage dimension for Internet monitoring.

Network incident lifecycle management. Given that cloud providers perceive most public Internet incidents as uncontrollable, they usually rely on rerouting mechanisms (e.g., changing PoPs) to bypass rather than patch issues. As a concrete example (Figure 4), once the primary AS paths start to experience link failures, the rerouting mechanism will shift traffic to backup network paths. However, the backup paths usually come with suboptimal performance, so we need to switch back to the primary paths when possible for better performance. As a result, after rerouting, we need to continuously monitor not only the in-use former backup paths to assure their health, but also the faulty primary paths so that we can switch back to the them once we detect that they are repaired This shows that *full lifecycle management that addresses both incident detection and repair detection* is an essential coverage dimension for Internet monitoring.

Per-tenant, upper-layer network metrics. Cloud providers

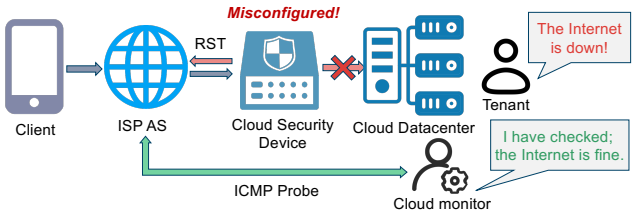


Figure 5: Example of upper layer incidents: RST reset

need to leverage a variety of network metrics for monitoring, which include not only L3 forwarding behavior, but also fine-grained signals carried by layer 4+ packet headers. For instance, we encountered cases where misconfigurations in security middle boxes triggered incorrect TCP resets (Figure 5), disrupting valid tenant TCP flows and causing connection terminations. In such scenarios, ICMP probe packets could still reach the client, making the issue invisible to coarse-grained reachability monitoring. Detecting this issue requires per-tenant, TCP session-level details to unveil the reset pattern. As another example, cloud tenants frequently blame the network for failures in their services, and the cloud provider must be able to validate/falsify such claims. This again requires per-tenant packet inspection as provenance. Consequently, *capturing both lower layer and per-tenant, upper layer network metrics* is yet another important coverage dimension that would enable algorithms for precise tenant-level incident detection and tenant report self-attestation.

2.3 Limitations of Existing Works

While cloud network monitoring has been a popular topic for many years, most existing discussions have focused on detecting and locating incidents within cloud datacenters and their private backbone network. Pingmesh [21] forms multiple levels of complete ping graphs across all cloud servers, enabling network problem diagnosis and troubleshooting. NetBouncer [51] leverages IP-in-IP encapsulation to actively probe cloud datacenter network paths, and employs algorithms to accurately localize device and link failures. However, these solutions cannot be directly extended to cover the public Internet, because their techniques were designed for the white-box datacenter environments with the assumptions of centralized management, controllable configurations, and known topology, dissimilar to the Internet environments.

Recent years have also seen the emergence of a few cloud-based public Internet monitoring systems. BlameIt [28] performs passive RTT measurements and coarsely classifies perceived incidents into cloud-side, AS path, or client-side. It then selectively trace-routes AS paths to localize the faulty AS. AAsclepius [52] mitigates Internet AS faults by monitoring, diagnosing, and circumventing them at the Cloud-Internet PoPs. The key innovation of AAsclepius is an approach to identify AS fault direction, so that they can bypass outbound (i.e., cloud-to-client) faults by detouring traffic through alternative AS paths. Note that this direction identification mechanism is orthogonal to the fine-grained bidirectional reachability discussed in this paper: the former aims to categorize known incidents, while the latter is in the context of detecting unknown incidents. At a high level, these existing works suffer from low coverage, as they rely on uni-directional (outbound) layer-3 (ICMP) probe packets to detect incidents (but not repairs). Consequently, they cannot address Internet incidents that require high coverage on bidirectional reachability, incident lifecycle, or tenant-aware metrics.

3 Skyline Overview

In this section, we discuss the system design challenges faced by Skyline, then walk though its system architecture.

3.1 System Design Challenges

Challenge 1: How to cover bidirectional reachability in an efficient and non-intrusive manner? A naive approach for covering bidirectional reachability is to extend existing outbound probing methods. However, given that each outbound probe can only cover the cloud subnet it originated from, full coverage essentially requires us to deploy probing agents in every cloud subnets, which is economically infeasible due to excessive host and EIP resource demands. An alternative solution is to design inbound probing originated from client side to monitor cloud subnet, but a strawman approach would require cloud tenants to respond to the probing signals, which is considered intrusive in public cloud environments, and raise compatibility, privacy, and governance concerns. Our solution to this dilemma is detailed in Section 4.1.

Challenge 2: How to steer probe packets among different paths flexibly? Before path recovery, we aim at probing the *exact* paths that the workload will take after the recovery—take outbound as an example, through the same PoP, out of the same ISP entrance, and targets the same client address space. Traditional monitoring falls short for two reasons. First, probe packets cannot be flexibly steered to the faulty AS path, because they by default follow the traffic engineering polices, thus sharing the same egress behavior with business traffic. Second, even if AS paths are fixed, outbound probes may still reach PoPs and ISP entrances via routes that differ from those used by business traffic, confounding assessment of the repair. Our solution is presented in Section 4.2.

Challenge 3: How to obtain per-tenant upper layer metrics efficiently and apply them to various use cases? Traditionally, per-tenant fine-grained network metrics are only visible at tenant hosts. However, collecting them directly from these hosts not only intrudes into tenant environments but also adds data path overhead. The first challenge is thus to develop collection methods that capture per-tenant L4-L7 network metrics efficiently and non-intrusively. After the collection, these metrics will be used for tenant-level incident detection and tenant report self-attestation, but these functionalities are sensitive to a wide variety of noise. The second challenge is therefore to carefully curate the system to achieve satisfying accuracy. We address these problems in 4.3.

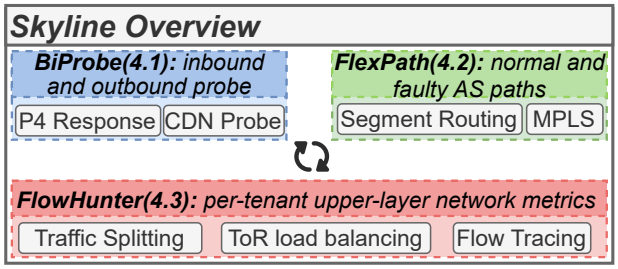


Figure 6: Skyline System Overview

3.2 Skyline Framework

Figure 6 shows the overall architecture of Skyline. It consists of three subsystems each address a design challenge.

BiProbe utilizes both outbound and inbound probing agents to cover fine-grained, bidirectional reachability of cloud subnets and client address spaces. Outbound reachability is covered by sending probe packets from cloud datacenters to client-side CDN nodes, while inbound reachability is covered by sending probe packets from client-side CDN nodes into P4-based cloud internet gateways (IGWs). This dual-agent design minimizes resource overhead by keeping most cloud subnets free of outbound probing hosts, and preserves transparency by having IGWs respond to inbound probes on behalf of cloud tenants. It enables deep visibility into asymmetric Internet failures that are hard to detect with uni-directional probing.

FlexPath employs segment routing with MPLS label binding to monitor both faulty and normal network paths simultaneously. It features a three-stage path steering mechanism, where the first stage direct probe packets to desired PoPs via dedicated cloud inter-datacenter tunnels, the second stage further steers the probe packets from PoPs onto interested ISP entrance, and the third stage sends the probe packets to representative client side IP addresses along AS paths. This mechanism enables full coverage over the incident lifecycle, allowing path recovery as soon as incidents are repaired.

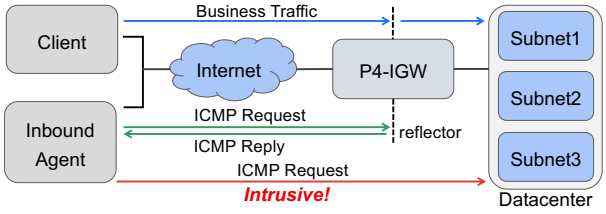
FlowHunter leverages an optical full-traffic replication system deployed at our PoPs. With the help of optical splitters that bypass tenant hosts, FlowHunter duplicates all tenant traffic metrics (e.g., TCP session information) without interfering with the data path, making the collection non-intrusive and achieving essentially zero performance overhead. The replicated traffic is then distributed to the analysis cluster. The cluster contains a scalable flow tracing system that realizes fine-grained visibility into tenant traffic, which facilitates accurate detection of tenant-level incidents and supports systematic self-attestation upon tenant reports.

4 System Design

In this section, we present the three subsystems of Skyline.

4.1 BiProbe: Bidirectional Probing

To address Challenge 1, we design BiProbe to cover bidirectional reachability in an efficient and non-intrusive manner.



src	dst	IGW	Explanation
inbound agent	subnet1	✅	P4 sends ICMP reply back to inbound agents
inbound agent	subnet2	❌	Inbound probe cannot reach IGW and subnet2
client	subnet1	✅	Regular traffic is forwarded to the tenants

Figure 7: Inbound probe with P4 IGW respond

Inbound probing with P4 response. As discussed in Section 3.1, sending inbound probe packets directly into cloud subnets is intrusive to tenants. We therefore need a way to test subnet reachability *without* entering cloud subnets. By examining the inbound traffic of our organization, we observe that all flows destined for cloud subnets first traverse cloud provider-operated IGWs built on Tofino P4 switches. These IGWs hold the forwarding state that maps external traffic to the appropriate cloud subnets. Our key observation is that, *if a packet can reach an IGW, it can also reach the cloud subnets associated with the IGW under normal cloud operation*. While there could be issues within the cloud providers, they can be detected independently by the existing datacenter monitoring mechanisms. This enables a tenant-bypassing inbound probing strategy: client-side agents send probes addressed to cloud subnets, while programmable IGWs intercept these probes and reply on the subnets' behalf, without forwarding them into the subnets themselves. The result is an inbound probing architecture that preserves tenant transparency and scales efficiently in our production environment.

Concretely, as shown in Figure 7, being a video-streaming company, we operate a large number of CDN nodes in populous client areas worldwide, which can be used as the inbound probing agents. We configure the match-action tables of the IGW programmable switches so that when packets originate from designated inbound agent IPs, the IGW swaps their source and destination fields, converts the packet type from ICMP *request* to *reply*, and sends the response back to the inbound agents directly, without forwarding probe packets into datacenter. Business traffic, in contrast, is normally forwarded by IGW to the tenant hosts through cloud network. This mechanism provides the same coverage to directly pinging every datacenter subnet, yet remains non-intrusive to production traffic. It is worth noting that ICMP probes are forwarded as normal transit traffic, thereby bypassing rate-limiting restrictions set by the ISP control plane. The pinglist strategy for inbound probing is outlined as follows: For each subnet with a mask of /24 or bigger, one destination IP is selected from each /24 ranges. For each subnet smaller than /24, a single destination IP is selected from their address space.

Outbound probing with CDN. In addition to covering reach-

Algorithm 1 BiProbe Outbound Probing Target Selection

```
// S_client, S_CDN denote set of client/CDN IPs in the region; t_s, t_p, t_l denote threshold on stablility, popularity, and pinglist size.
function PINGLISTGENERATION(S_client, S_cdn, t_s, t_p, t_l)
    for each IP ∈ S_cdn do
        stable_count[IP] ← 0, client_count[IP] ← 0
    for each probing round do
        PingList ← ∅ // Renew pinglist every round
        for each IP ∈ S_cdn do
            // Probe liveness of the CDN IP; IPs that are active across consecutive rounds are more stable.
            if Probing(IP) == Active then
                stable_count[IP] ← stable_count[IP] + 1
            else
                stable_count[IP] ← 0
            // Count the amount of clients the CDN IP serves
            client_count[IP] ← ClientUsage(IP, S_client)
        // Prioritize stable and popular CDN IPs
        S_stable ← {IP ∈ S_cdn | stable_count[IP] ≥ t_s}
        S_popular ← {IP ∈ S_stable | client_count[IP] ≥ t_p}
        // Sample CDN IPs to meet expected pinglist size
        PingList ← PrefixBucketing(S_popular, t_l)
```

ability of cloud subnets with inbound probing, BiProbe conducts outbound probing to assess client side reachability. The challenge is that we cannot directly send probe packets to client endpoints: On the one hand, direct probing is intrusive to the clients; on the other hand, the IP addresses associated with (often mobile) clients are highly unstable, yielding noisy results. To overcome these issues, BiProbe instead probes the CDN nodes used by clients. These CDN nodes provide stable, low-noise signals, allowing the system to reliably infer the reachability of the regions that clients reside in, and the reachability of the ASes that traffic to clients goes through. Crucially, driven by our video streaming business, we maintain an extensive global CDN deployment. This footprint is sufficient to characterize regional ISP outages that affect residing client endpoints, while assuring the monitoring is non-intrusive to the clients.

BiProbe also features an efficient, high-quality probing target selection and prioritization algorithm to generate the outbound pinglist. As shown in Algorithm 1, BiProbe updates the pinglist in every region once every round, where each round lasts three hours. At the beginning of each round, the system pings all candidate CDN IPs to test their liveness. For each IP, a *stable_count* tracks consecutive rounds of successful reachability, while a *client_count* records the number of clients currently served by that IP. CDN IPs with *stable_count* exceeding the stability threshold t_s are placed into a the set S_{stable}. Among them, the IPs that also satisfy the popularity threshold t_p further form the set $S_{popular}$, which represents CDN nodes that are most influential in terms of client coverage. The final pinglist for this round is then generated by applying the *Prefix Bucketing* algorithm to evenly select t_l CDN IPs from $S_{popular}$, so that probing targets are distributed

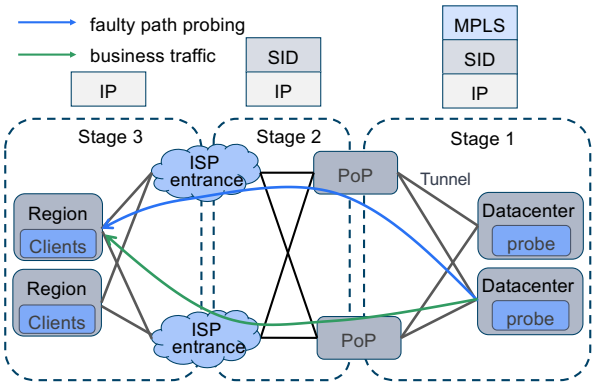


Figure 8: Workflow of FlexPath

across diverse IP prefixes, preventing local failures from being misinterpreted as widespread outages.

4.2 FlexPath: Flexible Path Steering

To address Challenge 2, we design FlexPath to steer probe packets onto the intended network paths. Our key observation is that probing path steering does not require control over every hop: accurate repair detection only demands that the probe and faulty traffic share the same tuple of cloud-controllable, path-defining hops—{cloud datacenter, PoP, ISP entrance, client address space}. Accordingly, rather than enforcing end-to-end control, FlexPath focuses on steering probe packets to the right PoPs, ISP entrances, and client side CDN nodes through three consecutive stages: datacenter-to-PoP tunneling, PoP-to-Internet steering, and Internet-to-client probing.

Stage 1: Datacenter-to-PoP tunneling. As shown in Figure 8, at the first stage, probe packets are carried from the cloud datacenter to a chosen PoP over the inter-datacenter backbone network. To prevent production traffic-engineering mechanism from deflecting probe packets, FlexPath uses a probe-dedicated controller that steers packets along deterministic tunnels. Concretely, the controller configures Linux kernels to tag probe packets with DSCP values via tc-flower, which map to static MPLS labels. This allows the probe packets to flexibly choose which backbone tunnels to go through, so that probe packets can examine the same tunnels that's going to be used after path recovery, regardless of whether they are active for production traffic at the moment. Assuring identical tunnels for faulty and recovered paths minimizes probing noises and improves accuracy of repair detection.

Stage 2: PoP-to-Internet steering. Upon reaching the backbone egress router associated with PoPs, probe packets are further directed to specific ISP entrance using segment routing. Before probing starts, the controller of FlexPath precomputes Segment Identifier (SID) stacks for each monitored ISP entrances, then instruments each probe packet with the SID stack towards the intended ISP entrance. This allows FlexPath to achieve deterministic PoP to ISP entrance control, unaffected by BGP routing decisions, ECMP load balancing, or transient failovers. It guarantees that each measurement

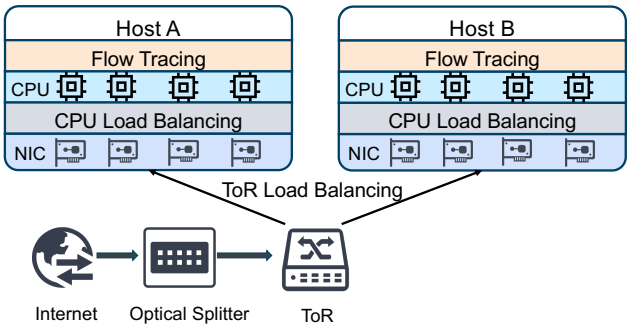


Figure 9: System architecture of FlowHunter

can be attributed precisely to a given ISP entrance, enabling fine-grained repair detection that would otherwise be masked by the nondeterminism of Internet routing.

Stage 3: Internet-to-client probing. FlexPath reuses the outbound probing pinglist of BiProbe generated by Algorithm 1 as the destinations of faulty path probe packets. This guarantees that the faulty path probe packets examine the status of the same ISP entrance to the same client side IPs and therefore the same AS paths experienced by the recovered production traffic, thus maximizing the fidelity of the probing results.

Life cycle management. Given the three-stage probing path control, FlexPath then manages the incident lifecycle according to the end-to-end path drop rate and RTT. If the drop rate of a faulty path remains below 0.5% and RTT fluctuations stay within 2% of the pre-fault baseline, we consider the path repaired and safely reroute production traffic to it. FlexPath always-on probing along designated end-to-end paths enables timely detection of repair events on faulty AS paths, allowing business traffic to be promptly shifted back to the primary path for better resource utilization and performance.

4.3 FlowHunter: Tenant Flow Collection

To address Challenge 3, we leverage FlowHunter to capture and analyze per-tenant upper-layer network metrics.

FlowHunter Architecture As shown in Figure 9, FlowHunter consists of the following three components: (1) Traffic Splitting: network traffic is duplicated via optical splitting to provide input for the analysis system. The optical splitter passively divides the optical power carried by a single input fiber into multiple output fibers with predetermined proportions, enabling efficient distribution of light signals used in passive optical networks. (2) Load Balancing: This operates on two levels. At the ToR switch level, traffic is divided across multiple servers by a load distribution system. Within each server, traffic is balanced across CPU cores, ensuring that traffic from the same tenant and TCP session is consistently handled by the same core. (3) Flow Tracing: Traffic are analyzed and aggregated into tenant-level metrics shown in Table 1, which include average RTT, 5-tuples, TCP flags (RST/SYN/FIN), the number of ICMP anomalies, and packet rate.

Tenant-level incident detection The collected metrics are dynamically compared against each tenant's historical baseline under normal conditions, enabling the identification of potential network anomalies. Specifically, we design a rule-based detection mechanism to identify abnormal behaviors and trigger alerts. If a tenant's average RTT increases significantly relative to its historical baseline and exceeds a predefined absolute threshold, it is considered an indication of degraded link performance. A sudden surge in TCP reset packets may suggest that intermediate devices or remote endpoints are actively terminating connections. A large number of "ICMP Unreachable" messages may indicate path drift, black hole routing, or packet loss on the Internet or cloud. A significant drop in PPS or BPS may signal path congestion or link instability. HTTP headers can be leveraged by Web Application Firewalls (WAFs) for application-layer attack and anomaly detection. Once an anomaly is detected, Skyline initiates a sequence of downstream actions. IPs of affected ASes are added to the pinglist of BiProbe, FlexPath initiates faulty path monitoring, and the associated flows are marked as unreliable to inform the scheduling system. Meanwhile, all anomaly events are logged for retrospective examination: For example, to provide data evidence to support tenants' claims of innocence after service issues arise.

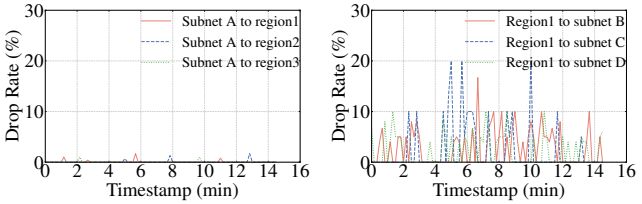
Tenant report self-attestation. In practice, the public cloud service chain is long and complex. When tenants experience service anomalies, the network is often the first suspected culprit. Whether it is delayed model training, connection interruptions, or system jitter, tenants tend to attribute responsibility to "poor Internet or cloud network quality." However, our analysis indicates that many user-perceived issues are not caused by public network links but instead stem from non-network factors such as misconfigurations, dependent service timeouts, link contention, and QoS constraints. To resolve accountability, FlowHunter provides an attestation filtering mechanism. By recording and analyzing full traffic data at PoPs, the cloud provider can promptly verify whether a tenant's network behavior was normal during an incident. We list the most representative filtering rules as follows:

- If no significant RTT increase or abnormal Reset/ICMP messages are observed during the anomalous period, this indicates that Internet did not degrade.
- If other tenants sharing the same path exhibit normal RTTs, the issue is likely confined to the application layer.
- If downstream traffic is delivered normally but no ACK responses are observed, this may suggest issues within the tenant's receiving stack.
- If there is a sudden surge in traffic without signs of congestion, it may point to insufficient flow control or a hot-start issue within the tenant's system.

This mechanism enables cloud providers to quickly assign responsibility during incident response, significantly improving the efficiency of fault localization. More importantly, tenants can access their own network snapshots as factual "proof of innocence," facilitating clear responsibility attribution among

Metric	Description	Usage
RTT	Client–gateway RTT (SYN+ACK)	Support RTT or jitter anomaly detection
5-Tuple	Src IP, Dst IP, Src Port, Dst Port, Protocol	Uniquely identify a flow and facilitate clustering analysis
TCP Flags	SYN, ACK, FIN, RST, etc.	Detect TCP-level abnormal connection disruptions
ICMP Signals	Destination Unreachable, TTL Expired, etc.	Indicate path changes or packet drops by intermediate nodes
PPS/BPS	Packets/Bytes per second	Detect throughput degradation or traffic bursts
Http Headers	Get, Post, Put, etc.	Support web application firewall

Table 1: Key metrics used in our system and their corresponding anomalous behaviors.



(a) Outbound probing result (b) Inbound probing result

Figure 10: Comparison of bidirectional reachability

the cloud, ISPs, and tenants. By enhancing transparency and trust, FlowHunter serves not only as a network monitoring system but also as a trusted recorder of network states between tenants and the cloud platform, supporting the construction of accountable and explainable cloud services.

Data path implementation. FlowHunter servers implement a CPU-level load balancer: Receive Side Scaling (RSS) and hash-based mapping are employed to ensure that packets belonging to the same TCP session are consistently processed by the same worker core, thereby preserving flow affinity and enabling accurate per-flow analysis. Each worker core is responsible for packet parsing, attribute extraction, feature accumulation, and lightweight preprocessing.

FlowHunter servers further adopt a customized DPDK pipeline for high-performance packet I/O: Two dedicated I/O cores continuously poll the NIC queues via Poll Mode Driver (PMD) and place incoming packets into the RX_RING. Subsequently, worker cores poll the RX_RING to aggregate per-tenant flow statistics, and write the aggregated results into the VIP_RING for downstream consumption. In the downstream analysis pipeline, a dedicated management core periodically consumes results from the VIP_RING and exports them to a distributed message queue (Kafka), where they are further processed by Flink for labeling, aggregation, and anomaly correlation. In parallel, specialized processes continuously compute upper-layer metrics such as RTT and TCP Reset counts, providing fine-grained visibility to support anomaly detection and tenant self-attestation mechanisms.

5 Evaluations

Skyline has been in production for over two years, deployed across all of our data centers to monitor traffic in and out of more than 40 PoPs and 200 ISP Entrances, providing comprehensive visibility into the Internet. In 2025, Skyline detected over 2,000 Internet incidents. To assess the precision (i.e. the fraction of system-reported incidents that are eventually confirmed) of Skyline, we manually processed all Skyline reported incidents and labeled it as either true positive or false positive. The results demonstrate that Skyline reaches a precision rate of ∼97%. To further assess the coverage (i.e., recall, the fraction of confirmed incidents that are reported by our system) of Skyline, we cross-referenced with all customer-reported incidents. If a confirmed incident is reported by our customer rather than Skyline, we label it as false negative. The results demonstrate that Skyline reaches a recall higher than 99%. Next, we evaluate the performance of Skyline in a large-scale data center, focusing on five key aspects: (1) effectiveness of BiProbe, (2) effectiveness of FlexPath, (3) effectiveness of FlowHunter, (4) detection precision, and (5) system overhead.

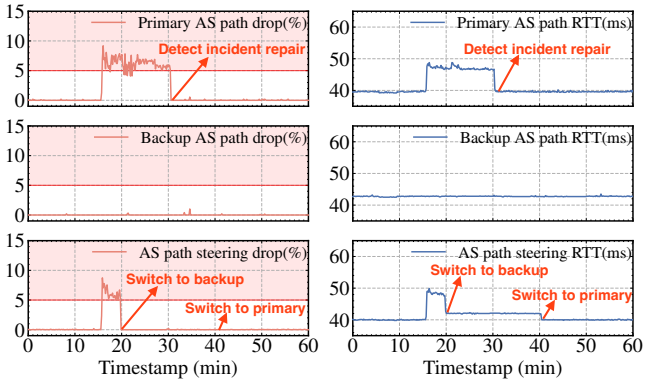
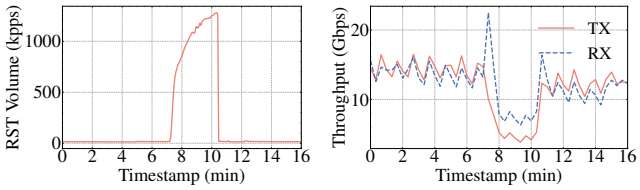


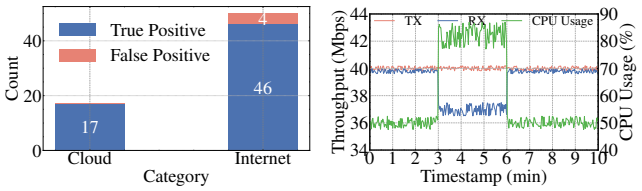
Figure 11: Incident lifecycle example. From top to bottom and left to right: (a) drop rate and (b) RTT of the primary AS path; (c) drop rate and (d) RTT of the backup AS path; (e) drop rate and (f) RTT of the business traffic. Metrics in (a)-(d) are measured by FlexPath, (e)–(f) are measured by BiProbe.



(a) Anomalous RST volume (b) Normal traffic throughput

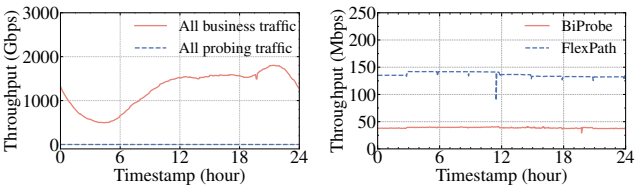
Figure 12: FlowHunter monitoring while RST case.

Effectiveness of BiProbe. As shown in Figure 10a, probers deployed in subnet A perform outbound probing toward Regions A, B, and C, showing near-zero packet loss. However, this does not mean that these regions can reach other cloud subnets. Figure 10b illustrates that probers deployed in Re-



(a) Precision of detection (b) Probe volume v.s CPU usage

Figure 13: Detection precision and false positive root cause.



(a) Business traffic throughput (b) Probing traffic throughput

Figure 14: BiProbe and FlexPath probing overhead

gion A perform inbound probing toward cloud subnets B, C, and D, revealing abnormal packet drop rate and indicating that these subnets are unreachable. This highlights the importance of covering every pairs of cloud subnets and client regions, underscoring the necessity of BiProbe which applies a combination of inbound and outbound probing to provide bindirectional coverage to Internet reachability.

Effectiveness of FlexPath. As shown in Figure 11, during an ISP line failure, FlexPath continuously monitored packet drop and RTT for the primary and backup AS paths, while BiProbe monitored the business traffic. At 16 minutes, both BiProbe and FlexPath detected over 5% packet drop rate (and abnormal RTT increase) on the primary path, indicating a line failure that necessitates path rerouting. Then at 20 minutes, business traffic was rerouted to the more stable backup path; BiProbe recorded packet drop reduction and RTT improvement, whereas FlexPath continued its monitoring on the faulty primary path. At 30 minutes, FlexPath detected that the primary path has been repaired, and started to signal that traffic can be switched back. Eventually, at 40 minutes, BiProbe observed RTT improvements, indicating that the business traffic was rerouted back to the primary path. Note that the 10-minute interval between repair detection and path recovery is set to assure the stability of the repaired path before rerouting. Overall, this case shows that FlexPath could detect incident repairs in real time, allowing our network to recover tens of minutes or even hours before official ISP repair notifications.

Effectiveness of FlowHunter. We use the FlowHunter per-tenant fine-grained metrics of the RST case to show the effectiveness. In March 2025, as shown in Figure 12a, FlowHunter detected a surge in anomalous RST volume at around 7 min. Meanwhile, Figure 12b shows a significant decline in tenants' business traffic, affecting both sending and receiving. After 5 minutes, a large number of tenants reported widespread network disconnections and we already prepare the following rerouting strategy. Upon investigation, the root cause was traced to a misconfiguration in security devices, which erroneously injected RST packets into a large amount of normal bussiness traffic. By tracing per-tenant, upper-layer network metrics, FlowHunter can detect incidents at tenant granularity, surpassing traditional Layer-3 monitoring approaches. Crucially, FlowHunter operates in a completely passive (bypass) mode, ensuring zero impact on tenant traffic.

Detection precision. We use one year of data from a single PoP to analyze the precision of Skyline. During this period, Skyline reported 67 incidents in the PoP, all within a single analysis interval (i.e., 2 minutes). As shown in Figure 13a, Skyline detected every Internet-related issue observed to date, including those originating from both ISP ASes and the cloud network. Of these, 73.02% were attributed to ISP ASes and 26.98% to the cloud network. Through manual assessment, we identify 63 of reported problems as true positives. Specifically, 46 of the 50 Internet issues were confirmed, while 4 were false positives. As shown in Figure 13b, the false positives arose from elevated CPU usage on the probe agents during a DDoS attack, which caused probe packet drops in the NIC receive stack as the host could no longer keep up with incoming ICMP replies. In subsequent deployments, we monitor the probe's CPU utilization to filter out such artifacts.

Skyline overhead. Figure 14a presents the overhead of all business traffic, bidirectional probing traffic, and FlexPath probing traffic at a single PoP. In Figure 14b, business traffic begins to decrease from midnight, reaching its minimum around 5:00, and then rises to its peak around 22:00 (the evening traffic peak), which aligns with business patterns; the average rate is 1,260 Gbps. Figures 8(b) and 8(c) show that both probing traffic volumes remain relatively stable. Due to differences in probing strategies, BiProbe has a higher probing frequency than FlexPath, with average traffic rates of 38.7 Mbps and 136.8 Mbps, respectively, accounting for only 0.0029% and 0.0109% of the total public Internet traffic. The associated overhead is therefore extremely low. FlowHunter employs passive optical splitting, ensuring zero overhead on business traffic. Its backend scales with PoP egress volume; for example, a representative cluster uses 4 mirroring switches to balance traffic across 27 servers, each equipped with 4×100G NICs to process ∼380 Gbps.

6 Experience

6.1 Network Incidents Found by Skyline

Table 2 classifies the root causes of Internet incidents found by Skyline into Internet-side and cloud-side issues, each further broken down into root cause categories that reflect common patterns observed in large-scale network deployments.

Internet-side issues. *ISP operational issues* are the most common source of disruptions (46.15%), arising from planned operational activities within of infrastructure. These include misconfigurations such as incorrect routing policies or overly permissive/strict filter rules, which can unintentionally block or leak prefixes; software upgrades introducing subtle con-

Side	Sub-Category	Root Causes
Internet	ISP operational issue	Configuration changes, software upgrades, topology adjustments, device replacement/migration
	ISP system issue	Device failures, line failures, software bugs, routing protocol failures, physical environment
	Network attack	DDoS, prefix hijacking/route leaks, reflection/amplification attacks
	Network Congestion	Peak hours at night, shopping festivals, major sport events, transit/peering bottlenecks
Cloud	Software bug	Bugs in routing resolution and distribution
	Hardware failure	Optical module failures, switch failures, board failures
	Cloud operational issue	Security appliance configuration errors

Table 2: Categorization of Public Internet Incident Root Causes

trol plane bugs, protocol incompatibilities, or unstable route convergence; topology adjustments like link reconfigurations, BGP community reassignments, or traffic engineering changes that inadvertently alter AS paths; and device replacements or migrations, such as swapping aging routers or relocating servers and switches across ASes. In practice, we locate the root causes of these issues by automatically ingesting and mapping ISP operational notifications into our system, then correlate them with the incidents detected by Skyline.

ISP system faults are disruptions originated from the devices themselves rather than operational activities, and account for 15.38% of the cases. In our deployment, we observed hardware failures such as line card or switching ASIC malfunctions, optical module degradation, and fiber disconnections causing intermittent packet loss. Firmware or protocol bugs including incorrect BGP route withdrawal handling or halted forwarding tables led to route oscillations and partial forwarding blackholes. Environmental factors, such as power instability or heat-induced transceiver degradation, further contributed to link flapping and silent drops. In practice, their root causes can only be obtained from ISP reports.

Network Attacks are incidents initiated by malicious attackers (7.69%). These include volumetric floods, protocol-level or application-layer attacks, and routing attacks such as prefix hijacking or route leaks, which can misdirect traffic and cause service interruptions. Reflection and amplification attacks further exploit open services (DNS, NTP, memcached) to magnify traffic volumes. In practice, once Skyline detects an incident, we diagnose these attacks through intrusion detection/prevention systems (IDS/IPS) together with traffic telemetry. *Network Congestion* (3.58%) are typically caused by higher volume during peak hours at night, shopping festivals, and major sport events, or bottlenecks on oversubscribed inter-ISP transit or peering links. Such congestion increases queuing delays, inflates RTT, and causes packet loss, thereby impairing service quality. Congestion within PoPs can often be localized with telemetry such as SNMP counters.

Cloud-side issues. Many Internet incidents (26.92%) originated from the cloud infrastructure then propagated outward to the Internet. For example, cloud control plane software bugs occasionally led to incorrect BGP announcements or inconsistent route visibility across data centers. Hardware faults, such as degraded optical modules, faulty NICs, or switch board failures, caused intermittent packet loss and throughput drops; in several cases, dust contamination or aging transceivers were

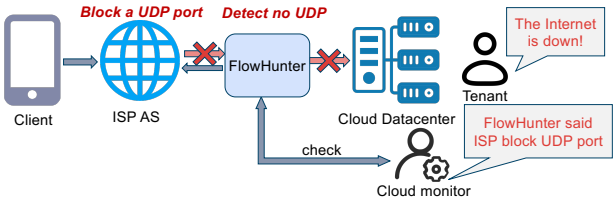


Figure 15: ISP block a UDP port case

the root cause. We also observed misconfigurations in security appliances and ACL policies, which silently dropped or misrouted tenant traffic. Since such incidents often occurred at PoPs which aggregate traffic from many tenants, they frequently caused large-scale service disruptions. In practice, these cases can be diagnosed with cloud datacenter monitoring systems together with logs from switches and hosts.

6.2 Real World Case Studies

We present representative incidents that Skyline uncovered in production, highlighting patterns that evade conventional monitoring but were successfully identified by Skyline.

ISP configuration changes introduced various routing issues. In October 2024, as shown in Figure 3, a metropolitan network cutover by a certain ISP resulted in the hijacking of two subnets belonging to a data center. Due to this event, BGP traffic destined for region A was mistakenly redirected to region B, causing 100% packet loss for two /21 subnets. We promptly detected the issue using BiProbe inbound probing, identified the scope of impact, and retrieved the affected tenant information through the FlowHunter system, enabling timely notifications to the respective tenants. Subsequently, leveraging reachability and performance information of other ASes from FlexPath, we advertised detailed /22 routes to various ISPs, successfully rerouting traffic to bypass the affected path. Other routing issues, such as ISP policy-induced routing loop that affect reachability across regions, can also be quickly detected via BiProbe inbound probing.

▷ **Experience 1:** BiProbe inbound probing can effectively detect routing issues caused by ISP configuration changes.

Internet attack induced large-scale packet loss. In July 2025, a volumetric attack on an ISP AS path caused severe congestion and packet loss, disrupting connectivity from a specific PoP to clients in multiple regions. BiProbe outbound probing detected the anomaly within one minute. Three minutes later, business traffic was resumed by isolating the faulty

ISP and rerouting via BGP, which successfully contained the impact. In contrast, the ISP itself did not detect the incident until nearly an hour later, at which point it notified the cloud provider. After 15 minutes, FlexPath detected that the incident was repaired, observing that the primary AS path's drop rate and RTT had returned to normal. After ten minutes of continuous assessment, traffic was safely rerouted to the primary AS path. The ISP formally confirmed incident repair roughly one and a half hours after the initial disruption.

▷ **Experience 2:** BiProbe incident detection and FlexPath incident repair detection both preceded official ISP notifications by tens of minutes or even hours.

Abnormal UDP blocking caused widespread service disruption. In July 2024, a large-scale service disruption occurred in a certain region due to ISP misconfigurations that blocked a range of UDP ports, lasting for approximately 6 minutes. During this period, 27.84% of tenant services that relied on UDP were impacted, with multiple tenant reporting issues including failed DNS resolution and unavailable media streaming. As shown in Figure 15, FlowHunter detected that UDP traffic to a specific port range was consistently dropped by the upstream ISP in real-time, while BiProbe outbound ICMP-based probing still signaled normal reachability to clients. After the misconfiguration was rolled back, it was concluded that stricter validation of ISP policy updates and fine-grained UDP-level traffic flow tracing are necessary to prevent such silent failures in the future.

▷ **Experience 3:** Internet incident detection must extend beyond ICMP to tenant-level, enabling timely identification of layer 4+ anomalies such as UDP blocking. FlowHunter flow tracing ability can detect upper-layer anomalies quickly.

Self-attestation with FlowHunter. In May 2024, a tenant reported persistent connection disruptions in a specific region. Although BiProbe added the client IP to the outbound pinglist and confirmed reachability, FlowHunter full-traffic flow tracing showed that downstream traffic was delivered but no ACKs were returned, causing repeated retransmissions and session stalls. This indicated anomalies beyond the network layer. Further tenant-side log and stack inspection revealed a misconfigured receiving stack that suppressed ACK generation, ultimately stalling TCP sessions. Beyond this case, we also observe issues such as sudden workload surges, delayed tenant scaling, and tenant software bugs, all of which can be captured through per-tenant upper-layer metric analysis of FlowHunter, thereby enabling self-attestation to tenants.

▷ **Experience 4:** FlowHunter enabled accurate tenant report self-attestation with upper-layer metrics, which helped distinguish between tenant-side issues and network-side failures.

An ISP's route update failure caused authorization issues. In November 2023, a network node adjusted its BGP routing policy, shifting the announcement of a certain IP prefix from one AS to another. However, the route registration was not updated, leaving the Route Origin Authorization (ROA) in the Resource Public Key Infrastructure (RPKI) still authorizing the original AS. As a result, the new AS was not considered a valid origin, and some overseas ISPs rejected the route based on RPKI validation, causing international connectivity issues and affecting access to domestic services from abroad. Through BiProbe's inbound probing, we detected the anomaly before significant service impact, identified the root cause as outdated route registration, and coordinated with the local Internet Network Information Center (INIC) to update the ROA, and eventually resolved the issue.

▷ **Experience 5:**With global RPKI adoption, BGP announcements must pass origin authentication or risk being filtered by strict ISPs. BiProbe Inbound probing can verify the BGP announcement authentication precisely.

6.3 Operational Lessons

Deployment steps. The evolution of Skyline reflects a continuous progression in our monitoring capabilities—expanding from uni-directional to bidirectional coverage, from incident detection to full lifecycle management, and from network-layer probing to tenant-level analytics. We began with standard outbound probing. While effective for basic ISP connectivity, this approach left cloud-side ingress blind spots. To improve bidirectional coverage, we advanced to BiProbe, introducing inbound probing via programmable IGWs to achieve agent-less monitoring for all cloud subnets, effectively detecting asymmetric routing failures. Subsequently, we observed that relying on ISP notifications for path recovery was operationally inefficient due to delays. We improved this by deploying FlexPath to continuously monitor faulty paths; this advanced our capability from passive waiting to proactive repair detection, significantly reducing recovery time. Finally, as tenant scale grew, distinguishing network faults from application issues became a bottleneck. To address this, we integrated FlowHunter to capture fine-grained L4-L7 metrics. This improvement transformed our diagnosis from simple connectivity checks to deep visibility, enabling definitive self-attestation and resolving ambiguity in tenant error reports.

BiProbe vs. FlowHunter. While overlapping with each other at first glance, these two subsystems actually address different parts of our problem and neither substitutes for the other. BiProbe does *active* reachability monitoring: it issues L3 probes to probe fine-grained, bidirectional reachability between client prefixes and cloud subnets. Because BiProbe operates independently of tenant traffic, it and can quickly flag path-level faults such as ISP line failures, leaks, or one-way hijacks. Crucially, when traffic is diverted before reaching the PoP, only BiProbe sees the loss of reachability; FlowHunter cannot observe flows that never arrive. FlowHunter does *passive* tenant-level monitoring: by duplicating business traffic, it surfaces L4–L7 symptoms (RTT inflation, retransmissions, RST spikes, UDP blocking) and provides per-tenant evidence for diagnosis and self-attestation. While passive signals can be noisy (e.g., client-side issues), they are indispensable for

incidents that do not manifest as simple reachability loss. In practice, A high-coverage monitoring system should leverage both BiProbe for rapid, active reachability issue detection, and FlowHunter for deep, passive tenant-level analysis.

Which ASes/regions need more attention. With tenant-level information from FlowHunter, we can precisely prioritize the ASes and geographic regions that require more attentions:

- *High-traffic, business-critical, and latency-sensitive regions:* Priority is given to areas hosting large-scale or core services, high-concurrency connections, or latency-sensitive workloads (e.g., real-time communication or high-frequency interactions), where disruptions can affect many users and cause significant business or user experience impact.
- *PoP depending on a single ISP:* When a PoP heavily depends on a single ISP, the lack of alternative egress options makes them frequent victims of widespread failures.
- *ISPs with a frequent failure history:* ISPs previously affected by disruptions due to routing loops, policy omissions, or uncontrolled changes should be continuously monitored.
- *ISPs with strict authentication mechanisms:* ISPs with strict RPKI validation, where route reachability is more susceptible to policy compliance issues.

6.4 Discussions and Future Work

Automated root cause diagnosis. While Skyline enables prompt detection and localization of Internet anomalies, pinpointing their root causes remains challenging if we rely solely on probing results. For example, persistent packet loss between two regions may be due to network congestion, failures in a specific ISP, or routing misconfigurations. Currently, operators must manually combine probing data with other telemetry, such as flow records and router logs, to determine the cause. A future direction is to integrate these data sources into Skyline and leverage automated analysis techniques (e.g., ML models) for rapid and accurate root cause diagnosis.

Impact-aware, cross-layer diagnosis. Skyline currently operates independently of application- and business-layer systems such as load balancers or traffic engineering, limiting its ability to correlate network anomalies with service-level symptoms. We plan to integrate Skyline with these components and combining tracing data from each to enable automated cross-layer analysis, distinguishing Internet incidents from application-level problems. This integration will also open doors to proactive remediation, such as traffic re-routing or automated failover, ultimately reducing both detection and resolution times in large-scale cloud environments.

Automatic rerouting. Skyline is primarily designed for large-scale monitoring and diagnosis, without built-in mechanisms for automatic rerouting. In the future, we plan to integrate Skyline with network control platforms (e.g., SDN controllers, traffic engineering systems) to enable closed-loop failure handling, where detection, impact assessment, and mitigation are executed in a coordinated, automated manner.

7 Related Works

Cloud network monitoring and diagnosis. A large body of work studies monitoring and diagnosing *provider-controlled* networks. Early systems advocate active probing to detect faults and performance regressions at scale [2, 3, 15, 17, 21, 51]. Complementary efforts focus on diagnosis and telemetry within datacenter clusters [22, 23, 25, 34, 41, 44], including systems that measure latency and localize failures using a predefined probing topology [21], capture packet-level evidence via switch-assisted filtering and tracing [57], steer probes onto specific internal paths to infer failed links/devices [51], and improve probe coverage with mirroring-guided and hybrid randomized/deterministic designs [12]. These approaches rely on strong administrative control and relatively stable topology inside the cloud; they do not directly extend to Internet-wide monitoring, where paths traverse black-box networks with rapidly changing routing and limited controllability.

Internet monitoring and diagnosis. Internet anomaly diagnosis spans three directions: (i) active approaches, which inject probes to detect anomalies [10, 37, 46, 55]; (ii) passive approaches, which analyze ongoing traffic without additional probing [8, 9, 29, 32, 33, 43, 47, 47, 56]; and (iii) hybrid approaches, which combine both methods above [28, 30, 52, 54]. Among them, BlameIt [28] and AAsclepius [52] are most closely related to our work. BlameIt first applies passive diagnosis to determine whether a fault lies in the Internet, then employs active diagnosis to pinpoint the specific faulty AS. AAsclepius automatically detects and detours Internet faults at the various PoP. However, both suffer from limited coverage dimensions and do not address the challenges faced by public cloud providers to enable self-attestation in Internet-scale monitoring and diagnosis. Espresso [53], Edge Fabric [48], Cascara [49], and AAsclepius [52] focus on traffic engineering and detouring, which are not the primary focus of our work and are left for future exploration.

8 Conclusion

Internet stability is vital to hyper-scale cloud providers, yet cloud-based monitoring system is lagging behind. Our key observation is that existing systems lack sufficient coverage across three distinct dimensions: fine-grained bidirectional reachability that covers both cloud subnets and client address spaces; incident lifecycle management that covers both incident detection and repair detection; and tenant-level analysis that covers upper layer network incidents and tenant self-attestation. In this paper, we introduce Skyline, which addresses each coverage dimension with one specialized subsystem Skyline has been deployed in our production environment and proven effective at identifying large amount of Internet incidents with high accuracy and low overhead.

ACKNOWLEDGMENTS

We thank our shepherd Mythili Vutukuru and the anonymous reviewers for their insightful comments and suggestions.

References

[1] J. Abbate. *Inventing the internet*. MIT press, 2000.

[2] B. Arzani, S. Ciraci, L. Chamon, Y. Zhu, H. H. Liu, J. Padhye, B. T. Loo, and G. Outhred. 007: Democratically finding the cause of packet drops. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 419–435, 2018.

[3] R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher. Pint: Probabilistic in-band network telemetry. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 662–680, 2020.

[4] D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, et al. {cISP}: A {Speed-of-Light} internet service provider. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 1115–1133, 2022.

[5] T. Bonald and L. Massoulié. Impact of fairness on internet performance. In *Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 82–91, 2001.

[6] M. Caesar and J. Rexford. Bgp routing policies in isp networks. *IEEE network*, 19(6):5–11, 2005.

[7] M. Calder, R. Gao, M. Schröder, R. Stewart, J. Padhye, R. Mahajan, G. Ananthanarayanan, and E. Katz-Bassett. Odin:{Microsoft's} scalable {Fault-Tolerant}{CDN} measurement system. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 501–517, 2018.

[8] D. R. Choffnes and F. E. Bustamante. Pitfalls for testbed evaluations of internet systems. *ACM SIGCOMM Computer Communication Review*, 40(2):43–50, 2010.

[9] D. R. Choffnes, F. E. Bustamante, and Z. Ge. Crowdsourcing service-level network event monitoring. In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 387–398, 2010.

[10] Í. Cunha, P. Marchetta, M. Calder, Y.-C. Chiu, B. V. Machado, A. Pescapè, V. Giotsas, H. V. Madhyastha, and E. Katz-Bassett. Sibyl: a practical internet route oracle. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 325–344, 2016.

[11] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. Luckie, R. K. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and K. Claffy. Inferring persistent interdomain congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 1–15, 2018.

[12] R. Ding, X. Liu, S. Yang, Q. Huang, B. Xie, R. Sun, Z. Zhang, and B. Cui. Rd-probe: Scalable monitoring with sufficient coverage in complex datacenter networks. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 258–273, 2024.

[13] H. L. Dreyfus. *On the internet*. Routledge, 2008.

[14] R. Fanou, F. Valera, and A. Dhamdhere. Investigating the causes of congestion on the african ixp substrate. In *Proceedings of the 2017 Internet Measurement Conference*, pages 57–63, 2017.

[15] S. K. Fayaz, T. Sharma, A. Fogel, R. Mahajan, T. Millstein, V. Sekar, and G. Varghese. Efficient network reachability analysis using a succinct control plane representation. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 217–232, 2016.

[16] N. Feamster and H. Balakrishnan. Detecting bgp configuration faults with static analysis. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 43–56, 2005.

[17] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum, and A. Vahdat. {SIMON}: A simple and scalable method for sensing, inference and measurement in data center networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 549–564, 2019.

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

[19] G. Giuliari, T. Klenze, M. Legner, D. Basin, A. Perrig, and A. Singla. Internet backbones in space. *ACM SIGCOMM Computer Communication Review*, 50(1):25–37, 2020.

[20] A. N. Group. Evaluation methods and best practices for public network quality in cloud computing. *Alibaba Cloud Blog*, 2024. https://developer.aliyun.com/article/783391.

[21] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, et al. Pingmesh: A large-scale system for data center network latency measurement and analysis. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 139–152, 2015.

[22] A. Gupta, R. Harrison, M. Canini, N. Feamster, J. Rexford, and W. Willinger. Sonata: Query-driven streaming network telemetry. In *Proceedings of the 2018 conference of the ACM special interest group on data communication*, pages 357–371, 2018.

[23] N. Handigol, B. Heller, V. Jeyakumar, D. Mazières, and N. McKeown. I know what your packet did last hop: Using packet histories to troubleshoot networks. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 71–85, 2014.

[24] T. Holterbach, E. C. Molero, M. Apostolaki, A. Dainotti, S. Vissicchio, and L. Vanbever. Blink: Fast connectivity recovery entirely in the data plane. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 161–176, 2019.

[25] Q. Huang, H. Sun, P. P. Lee, W. Bai, F. Zhu, and Y. Bao. Omnimon: Re-architecting network telemetry with resource efficiency and full accuracy. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 404–421, 2020.

[26] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an ip backbone. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 237–242, 2002.

[27] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. Cooperative content distribution and traffic engineering in an isp network. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 239–250, 2009.

[28] Y. Jin, S. Renganathan, G. Ananthanarayanan, J. Jiang, V. N. Padmanabhan, M. Schroder, M. Calder, and A. Krishnamurthy. Zooming in on wide-area latencies to a global cloud provider. In *Proceedings of the ACM special interest group on data communication*, pages 104–116. 2019.

[29] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 246–259, 2010.

[30] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving beyond end-to-end path information to optimize cdn performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 190–201, 2009.

[31] A. Lakhina, M. Crovella, and C. Diot. Characterization of network-wide anomalies in traffic flows. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 201–206, 2004.

[32] R. Landa, L. Saino, L. Buytenhek, and J. T. Araújo. Staying alive: Connection path reselection at the edge. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 233–251, 2021.

[33] R. Landa, L. Saino, L. Buytenhek, and J. T. Araújo. Staying alive: Connection path reselection at the edge. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 233–251, 2021.

[34] J. Langlet, R. Ben Basat, G. Oliaro, M. Mitzenmacher, M. Yu, and G. Antichi. Direct telemetry access. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 832–849, 2023.

[35] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff. A brief history of the internet. *ACM SIGCOMM computer communication review*, 39(5):22–31, 2009.

[36] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, and K. Claffy. Challenges in inferring internet interdomain congestion. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 15–22, 2014.

[37] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 367–380, 2006.

[38] R. Mahajan, D. Wetherall, and T. Anderson. Understanding bgp misconfiguration. *ACM SIGCOMM Computer Communication Review*, 32(4):3–16, 2002.

[39] A. A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards automated performance diagnosis in a large iptv network. *ACM SIGCOMM Computer Communication Review*, 39(4):231–242, 2009.

[40] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot. Characterization of failures in an operational ip backbone network. *IEEE/ACM transactions on networking*, 16(4):749–762, 2008.

[41] M. Moshref, M. Yu, R. Govindan, and A. Vahdat. Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 129–143, 2016.

[42] W. B. Norton. Internet service providers and peering. In *Proceedings of NANOG*, volume 19, pages 1–17, 2001.

[43] V. N. Padmanabhan, S. Ramabhadran, and J. Padhye. Netprofiler: Profiling wide-area networks using peer cooperation. In *International Workshop on Peer-to-Peer Systems*, pages 80–92. Springer, 2005.

[44] Y. Peng, J. Yang, C. Wu, C. Guo, C. Hu, and Z. Li. {deTector}: a topology-aware monitoring system for data center networks. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 55–68, 2017.

[45] M. E. Porter and M. ilustraciones Gibbs. Strategy and the internet. *Harvard business review*, 79(3):62–79, 2001.

[46] L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding internet reliability through adaptive probing. *ACM SIGCOMM Computer Communication Review*, 43(4):255–266, 2013.

[47] B. Schlinker, I. Cunha, Y.-C. Chiu, S. Sundaresan, and E. Katz-Bassett. Internet performance from facebook's edge. In *Proceedings of the Internet Measurement Conference*, pages 179–194, 2019.

[48] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. Engineering egress with edge fabric: Steering oceans of content to the world. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 418–431, 2017.

[49] R. Singh, S. Agarwal, M. Calder, and P. Bahl. Cost-effective cloud edge traffic engineering with cascara. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 201–216, 2021.

[50] S. Sundaresan, W. De Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: a view from the gateway. *ACM SIGCOMM computer communication review*, 41(4):134–145, 2011.

[51] C. Tan, Z. Jin, C. Guo, T. Zhang, H. Wu, K. Deng, D. Bi, and D. Xiang. {NetBouncer}: Active device and link failure localization in data center networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 599–614, 2019.

[52] K. Yang, Y. Li, S. Long, T. Yang, R. Miao, Y. Zhao, C. Ji, P. Mi, G. Yang, Q. Xie, et al. {AAsclepius}: Monitoring, diagnosing, and detouring at the internet peering edge. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 655–671, 2023.

[53] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, et al. Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *Proceedings of*

the Conference of the ACM Special Interest Group on Data Communication, pages 432–445, 2017.

[54] M. Zhang, C. Zhang, V. S. Pai, L. L. Peterson, and R. Y. Wang. Planetseer: Internet path failure monitoring and characterization in wide-area services. In *OSDI*, volume 4, pages 12–12, 2004.

[55] Z. Zhang, M. Zhang, A. G. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian. Optimizing cost and performance in online service provider networks. In *NSDI*, pages 33–48, 2010.

[56] Z. Zhang, M. Zhang, A. G. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian. Optimizing cost and performance in online service provider networks. In *NSDI*, pages 33–48, 2010.

[57] Y. Zhu, N. Kang, J. Cao, A. Greenberg, G. Lu, R. Mahajan, D. Maltz, L. Yuan, M. Zhang, B. Y. Zhao, et al. Packet-level telemetry in large datacenter networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 479–491, 2015.