HERMES: Repurposing User-Driven Speed Tests to Monitor the Internet

Loqman Salamatian*
Columbia University
New York, NY, USA

Kevin Vermeulen
LIX-CNRS
Palaiseau, France

Dave Choffnes
Northeastern University
Boston, MA, USA

Ethan Katz-Bassett
Columbia University and Google
New York, NY, USA

Phillipa Gill
Google
New York, NY, USA

Abstract

Diagnosing performance degradations and pinpointing their source is crucial for operators to make informed routing decisions and for policymakers and researchers to assess the Internet's stability, yet no publicly available observatories currently provide this capability. Existing solutions rely on coarse-grained signals that fail to capture end-user performance, while proprietary solutions are inaccessible and offer limited attribution for identifying the source of a problem. We introduce HERMES, the first *open* system to fill this gap. HERMES uses publicly available M-Lab speed tests—data that has existed for years but has not previously been used to automatically detect and explain end-user performance degradations at scale. To achieve these goals, HERMES combines statistical techniques to detect performance degradation with novel tomography methods and forward and reverse path measurements to localize the source of a problem. Despite relying only on public data, HERMES matches a reimplementation of a large cloud provider's monitoring system for 94.5% of events visible to both systems, agreeing on the degradation source in the path. HERMES also surfaces 11× more publicly discussed events than existing public observatories. We demonstrate its ability to track weather- and cable-cut disruptions, diagnose routing inefficiencies, and identify persistently congested links.

CCS Concepts

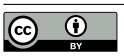
• **Networks** → **Network measurement**; **Network performance analysis**.

Keywords

Internet measurement, performance monitoring, network tomography, anomaly detection, speed tests, M-Lab

ACM Reference Format:
Loqman Salamatian, Kevin Vermeulen, Dave Choffnes, Ethan Katz-Bassett, and Phillipa Gill. 2026. HERMES: Repurposing User-Driven Speed Tests to Monitor the Internet. In *ACM SIGCOMM 2026 Conference (SIGCOMM '26), August 17–21, 2026, Denver, CO, USA.* ACM, New York, NY, USA, 24 pages. https://doi.org/10.1145/3789240.3829129

*Work initiated during a Google internship.



SIGCOMM '26, Denver, CO, USA

ACM ISBN 979-8-4007-2467-1/2026/08
https://doi.org/10.1145/3789240.3829129

1 Introduction

The Internet is a critical infrastructure whose performance directly affects user-facing applications, including interactive services such as gaming and video conferencing, where increases in latency or packet loss can significantly degrade user experience. However, existing public Internet observatories mainly focus on outages, shutdowns, and censorship [1, 19, 42, 70], whereas sustained performance degradations (e.g., persistent increases in latency or throughput drops short of outages) are primarily visible through private monitoring by cloud providers and third parties (§2). This lack of shared visibility limits transparency and prevents operators and researchers from understanding the scope and impact of performance problems.

Performance degradations should be observable in the same way outages are. Public, third-party measurements provide a shared evidence base that supports diagnosis, postmortems, and coordination across organizational boundaries without relying on proprietary telemetry. From our experience working with Content Delivery Networks (CDNs) and Internet Service Providers (ISPs), resolving performance problems often requires cross-network collaboration, which is difficult in the absence of shared, publicly verifiable measurements. Improved visibility can also inform broadband funding (*e.g.,* BEAD [65]).

Repurposing speed test data to detect performance degradation: Our starting insight is that the Network Diagnostic Tool (NDT) speed test data [60] can be repurposed to systematically identify performance degradations and, when combined with topology measurements, localize where they occur. The test is directly accessible from Google's search results, making it one of the most widely used Internet performance measurements, with approximately 4 million tests conducted daily across diverse locations and networks. M-Lab publishes the test data daily (§5.2). Unlike other speed test platforms that measure performance within the user's ISP network [18, 56], NDT speed tests specifically assess performance across peering interconnections, frequent points of congestion [25, 29, 59]. Importantly, most tests are user-initiated and therefore tend to occur when users are motivated to assess their connectivity, meaning the dataset is biased toward periods when network performance is salient to end users. NDT pairs tests with traceroutes and reverse traceroutes, providing bidirectional visibility into network paths [97]. This combination of performance data and bidirectional path visibility enables end-to-end topology mapping and localization of user-impacting degradations (§2).

Despite its breadth, this dataset has been underutilized for systematic, large-scale event detection. By *event detection*, we mean automated identification of statistically significant degradations in network performance (e.g., increased latency) that persist long enough to merit manual investigation and affect groups of users across a shared network or location. Prior research using M-Lab data has largely been used to study specific phenomena, such as congestion at interconnection points [59], users' Internet plans [75] and accessibility [68, 74], Starlink's network performance evolution [64], or Venezuela's political crisis [13]. While valuable, these studies require researchers to select events, hypothesize their effects, and craft custom analyses, making them impractical for continuous, automated monitoring.

Three challenges hinder automated detection from M-Lab data: (i) speed tests reflect localized performance and may be confounded by customer-side issues, complicating the separation of last-mile effects from broader network disruptions; (ii) tests are user-triggered, leading to irregular and uneven sampling; and (iii) test participants may not be representative of overall Internet conditions, raising concerns about coverage and bias.

Overview. HERMES detects performance degradations using statistical hypothesis testing and distributional comparisons to identify significant shifts in latency or throughput (§4.1). HERMES localizes their sources using a novel tomography-inspired method over bidirectional paths, distinguishing routing-induced degradation from congestion on stable routes (§§ 4.2 and 4.3). When the source remains ambiguous, HERMES triggers additional targeted measurements to disambiguate competing explanations (§4.4). HERMES operates at a daily granularity because M-Lab publishes its speed tests once a day; this is also a reasonable default because it emphasizes more persistent degradations that are more likely to reflect sustained user impact and warrant operational follow-up.

Over a 5-month period, HERMES has processed 4 billion speed tests, detected 65K network events affecting 37K ⟨AS, metro⟩[1] pairs across 9,710 ASes in 166 countries, only 5.2% of which were detected by other public observatories. Compared to BlameIt [47], a tool developed by a major cloud provider that we reimplement within AnonCDN, HERMES achieves similar precision (§5.1); compared to existing public observatories, HERMES detects 12× more publicly discussed events (§5.2). We evaluate HERMES's geographic and network reach and find that, for user traffic traversing paths to M-Lab servers, HERMES achieves sufficient coverage to detect degradations affecting between 60-80% of Internet users in Europe and North America, and between 40-70% in Africa (§5.3). Its use of bidirectional path measurements is key to accurately diagnosing network issues (§5.4). Operationally, HERMES can surface likely sources of user slowdowns that may inform traffic engineering decisions, swifter escalation to peers and postmortem analysis. We are continuing to run HERMES as an ongoing public observatory and share its detected events [63].

Our work is in line with the community's ethical standards. We use de-identified, public speed tests from M-Lab, and we analyze data in aggregate without any attempt to reidentify participants. Before users trigger a speed test, they are presented with text informing them that their IP address and test results will be collected by M-Lab, publicly released, and used for Internet research, in accordance with M-Lab's privacy policy [62].

[1]Here metro means metropolitan area.

2 Related Work

Academic efforts focused on liveness signals and congestion detection: These approaches rely on BGP updates, active probing (*e.g.,* ping, traceroute), or unsolicited traffic observed with telescopes [8, 26, 30, 42, 49, 52, 57, 79, 100]. BGP provides coarse visibility, but most announcements reflect routine churn [40], and transient congestion often leaves no control-plane trace (§3). Active probing can provide path information and confirm reachability but systems, such as IODA [42] and 007 [4], focus on detecting outages and responsiveness rather than end-user performance degradation. In contrast, HERMES uses measurements from M-Lab servers of application-level performance paired with forward and reverse traceroutes, enabling visibility into performance and bidirectional paths. Prior work highlights the ambiguity of inferring interdomain congestion from end-to-end measurements. Jitterbug [14] derives round trip time (RTT) metrics from jitter dispersion but depends on dense, continuous probing. Luckie et al. [53] emphasize challenges from alias resolution, router ownership, and path asymmetry; Sundaresan et al. [95] extend these concerns and show that common throughput-inference assumptions on M-Lab data often fail and note biases from crowdsourced sampling and home-network variability. While the accuracy of NDT for estimating absolute bandwidth has been questioned [56] and its methodology has evolved (e.g., MSAK [61]), these issues are tangential to our approach: HERMES uses throughput only as a *relative* performance signal. By comparing temporal shifts within stable user groups (defined by shared ⟨AS, metro⟩, with additional constraints in Section 4.1.1), HERMES detects degradations even when absolute test values fluctuate.

Privileged performance monitoring: A second category of related work uses privileged vantage points operated by cloud providers, CDNs, or commercial monitoring platforms. These systems have access to dense measurements, controlled endpoints and global traffic visibility, which enables them to detect and localize problems using simple statistical methods. However, their visibility is centered on their own services and infrastructure: most providers cannot instrument arbitrary clients at Internet scale, and paths to caches or cloud endpoints may differ from paths to ordinary clients. Their localization also typically stops at the AS level, and the data is proprietary and closed. By contrast, HERMES enables finer-grained Internet tomography using only open, user-driven measurements. Despite being sparse, noisy, and unevenly distributed, these data allow HERMES to attribute degradations not only to entire networks but also to specific metropolitan areas or peering links. This capability supports detection of a broader range of events, from network-wide outages to metro-level disruptions and single-interconnection congestion, and requires statistical inference techniques that go beyond those used in privileged-vantage-point systems. BlameIt [47], originally deployed at Microsoft, localizes issues at the AS level using a massive amount of regularly scheduled traceroutes and measurements of end-to-end user connections to Microsoft services. The approach relies on the assumption that latency noise

Table 1: Comparison of HERMES with other approaches in terms of key features, including application-level metrics, the ability to pinpoint causes, insights into forward and reverse paths, open data availability, end-user perspective, and data plane capabilities. HERMES uniquely satisfies all listed properties.

Property	HERMES	IODA [42]	PlanetSeer [100]	007 [4]	BlameIt [47]	Skyline [36]	Cloudflare Radar [19]	CEM [17]
End-User Perspective	✓	x	✓	x	✓	✓	✓	✓
Performance Metrics	✓	x	x	x	✓	✓	✓	✓
Pinpoint Causes	✓	x	✓	✓	✓	x	x	✓
Bidirectional Paths	✓	x	✓	x	x	✓	x	✓
Open Data	✓	✓	x	✓	x	x	x	x

can be smoothed out through frequent, uniformly distributed probes—a condition that holds in cloud-scale deployments but not when measurements are unevenly distributed and too sparse for such averaging. Contemporaneous with HERMES, Skyline [36] extends the cloud-centric approach with bidirectional probing and path-steered repair monitoring. Skyline uses the cloud's control over both probing endpoints to diagnose and repair paths between a cloud site and an ISP-hosted cache, whereas HERMES uses public measurements from end users to localize degradations on ordinary Internet paths without controlling either endpoint. Earlier systems such as PlanetSeer [100] and CEM [17] also used application-facing measurements to detect network events. PlanetSeer monitored path failures affecting PlanetLab-hosted services, while CEM used BitTorrent activity to expose network events; both settings have since disappeared or declined in relevance. HERMES follows this same principle of repurposing available measurements, but applies it to open user-driven speed tests.

Partially public but coarse observability: A third category of systems provides public visibility but lacks either path-level attribution or direct performance measurements. Cloudflare Radar [19] identifies outages by analyzing aggregate traffic volumes but does not expose the path information needed to localize the network source of end-user degradations. Ookla's Open Data [69] and Cloudflare Speed Test [21] provide performance data only at coarse geolocation granularity and without the bidirectional path measurements needed for tomography. Crowdsourced incident platforms such as Downdetector [27] and IsItDownRightNow [45] rely on user-submitted reports and social media activity to detect outages. While useful for surfacing disruptions quickly, these platforms do not run network measurements, and their detection methodologies are neither transparent nor peer-reviewed. HERMES uses open data, combining end-user performance signals with forward and reverse path measurements to localize user-facing degradations.

Table 1 synthesizes the systems discussed in this section. No prior system offers open, path-aware, performance-focused observability at Internet scale; HERMES fills this gap.

3 An Illustrative Example

We illustrate HERMES with a performance incident between Cogent (AS174) and TATA (AS6453) observed in our dataset. This event went undetected by public observatories (e.g., Cloudflare Radar, IODA [20, 43]) but drew over 100 Reddit comments [80]. This example demonstrates the challenges HERMES is designed to address and previews how our methodology overcomes them.

Figure 1 shows round-trip latency and throughput measurements from NDT speed tests between Cogent users in Chicago and an M-Lab server hosted in Chicago by TATA. On July 4th, the latency increased and the throughput sharply decreased, signaling a performance degradation. A single speed test would only suggest degraded performance for an individual user. On its own, such a result provides no evidence that the problem generalizes to other users; in fact, even under normal conditions, throughput measurements include a mix of high and low-values for reasons unrelated to network issues (e.g., Wi-Fi issues, background traffic). By contrast, examining multiple speed tests from many users served by the same network in the same location to the same destination reveals a simultaneous latency increase and throughput decrease—suggesting that the problem is deeper in the network path.

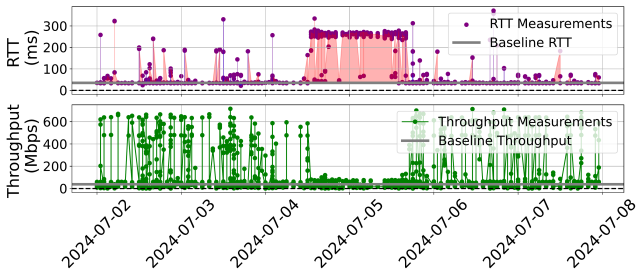


Figure 1: Round-trip latency (top) and throughput (bottom) between Cogent users in Chicago and a TATA-hosted M-Lab server in Chicago during July 2-7, 2024. Baselines correspond to median values over the preceding week.

While these aggregated performance metrics reveal a widespread event, they do not explain its source. To localize the problem, we correlate performance signals with topology data derived from bidirectional traceroutes that are run alongside the speed tests. The forward direction, from the M-Lab server to the users, reveals that although the AS-level path between Cogent and TATA had remained stable (*i.e.,* direct interconnection), the geographic path (Figure 2) was rerouted through a more distant interconnection point in Texas. Still, this explanation alone falls short of accounting for the magnitude of the observed latency increase (more than 210 ms observed versus ≈ 30 ms incurred by detouring through Texas). Adding reverse-path measurements, from the users to the M-Lab server, reveal detour on the reverse paths stretching as far as Singapore. Combined, the detours in both directions explain approximately 190 ms of the observed latency. Furthermore, no traceroutes traversed the Chicago/NYC peering points that day, indicating that they might have gone down and traffic had to rely on far-flung fallback. The study of the reverse paths alongside the performance metrics reveals an event of much greater severity and complexity than one could infer from forward-path alone.

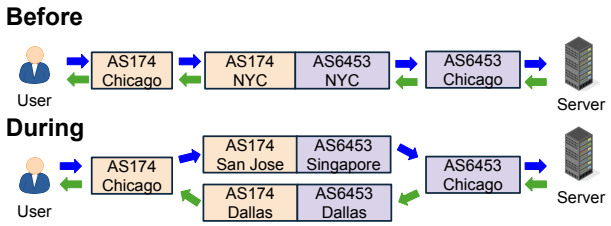


Figure 2: Bidirectional path changes during the incident. Before the event, traffic traverses Cogent and TATA in Chicago and NYC; during the event, paths detour via Dallas, San Jose, and Singapore.

This example points to two key goals of HERMES: detecting performance degradation and identifying the network entities responsible—whether by origin or by effect. In this case, the primary trigger appears to be the unavailability of peering links between Cogent and TATA in Chicago and NYC. However, the resulting detours through Dallas and San Jose, then through Singapore are themselves responsible for the bulk of the observed latency increase. HERMES aims to identify both: the disruption (e.g., unavailable direct interconnection) and the degraded alternative path (e.g., long detour through Asia) that together explain the observed user experience. We refer to both classes of entities as *sources* of an event and design our methodology to surface them through joint analysis of performance and topology. Our findings align with the reasons mentioned on the Cogent Status webpage, which point to a nationwide incident affecting Cogent and specific issues in New Jersey [80].

4 Methodology

The example above highlights core challenges that shape the design of HERMES, leading to three design questions:

How can we determine if we have adequate spatial and temporal coverage to reliably detect events? One of the first hurdles HERMES faces is the uneven distribution of speed tests, which results in gaps in spatial and temporal coverage and biases due to the tendency for users to issue measurements when experiencing problems. Rather than attempting to overcome this bias, HERMES uses it to its advantage: the skew toward problem-driven measurements aligns with our goal of detecting as many network events as possible (Appendix B.7 verifies this assumption post-hoc). We detail how HERMES aggregates and interprets measurements to make statistically robust inferences (§4.1).

How do we identify the sources of events? The second hurdle we faced is that performance degradation can stem from two different situations: (i) path changes, or (ii) deteriorating conditions (e.g., congestion) along the same path. Our methodology operates on the assumption that either case must be caused by an entity along the user's path or by a change in the path itself—even if that change was originally triggered by events occurring elsewhere in the network, as highlighted by Poiroot [46]. To distinguish these scenarios and accurately isolate sources, HERMES constructs a topology (§4.2) and applies a two-pronged approach (§4.3): (1) *Temporal Tomography*, which identifies entities whose involvement in anomalous paths changes significantly during the event compared to prior days; and (2) *Correlation Tomography*, which surfaces entities that appear across the anomalous paths of many affected user groups, indicating a likely shared source of disruption. By combining these two forms

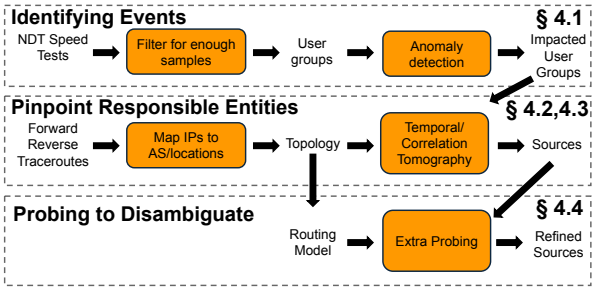


Figure 3: Schematic representation of HERMES.

of evidence, HERMES identifies entities responsible for an event, even in the face of limited visibility and sparse measurement coverage.

How do we resolve ambiguity about responsible network entities? Even with correlated performance and topology data, some issues remain ambiguous. Multiple network paths may overlap in complex ways, making isolating the exact networks or links responsible for an event difficult. To resolve these ambiguities, HERMES issues additional targeted regular and reverse traceroutes. The intuition is simple: a single well-placed probe can act like a spotlight, confirming one possible explanation while ruling out others. By greedily directing these probes toward user groups and paths that would reduce the most uncertainty, HERMES allocates scarce measurement capacity where it matters most (§4.4).

Putting it all together. HERMES combines user-driven data, topology inference, and targeted probing into a unified methodology that transforms NDT tests into Internet-wide insights. Figure 3 shows the pipeline: tests are grouped by user groups (§4.1.1), analyzed for anomalies (§4.1), mapped onto a bidirectional traceroute-derived topology (§4.2), localized with temporal and correlation tomography (§4.3), and refined through targeted probing (§4.4).

4.1 Identifying Events

HERMES identifies *events* as statistically significant degradations in latency or throughput relative to a historical baseline for a given user group. Speed-test data is well suited to this task because it directly measures user-experienced performance and because users often initiate tests *in response to* poor experience, introducing a positive selection bias that concentrates measurements around real degradations [9]. By default, HERMES operates at a daily granularity. This choice matches M-Lab's data publication cadence and provides sufficient sample sizes for robust inference. Daily aggregation also avoids a key confounder in sub-daily detection: Internet performance exhibits pronounced diurnal patterns, so comparing an individual hour against a day-level baseline can mistake normal peak/off-peak variation for a degradation. Hour-aware baselines address this issue, but at the cost of coverage. Each hour must independently satisfy the minimum sample and unique client requirements, so many user groups that are well covered at the daily level become too sparse when split into finer bins. In Appendix A.4, we evaluate a proof-of-concept sub-daily variant of HERMES that constructs hour-aware baselines and tests 1, 6, and 12 hour windows. The results expose a coverage–resolution trade-off. With hourly bins and a 14-day lookback, among user groups that satisfy the daily coverage requirements, fewer than 25% have enough measurements in more than half of the 24 hourly bins, and fewer

than 10% have enough measurements in all 24 hours. Intermediate windows, especially 6-hour bins, recover the largest fraction of events that are invisible at daily granularity while retaining more usable coverage. These additional events are typically short-lived, spanning only a small number of bins.

We first establish a baseline representing normal operation for each metric—throughput, latency. This requires grouping users into User Groups (§4.1.1), based on shared network and geographic characteristics. Once these groups are defined, we compute baseline values for each metric at a daily granularity to account for natural variations in performance (e.g., diurnal load patterns). The next step is to identify when a user group significantly deviates from its baseline. This process is conducted independently for each metric and requires statistically robust tools to ensure that deviations cannot be explained by regular fluctuations but instead reflect true performance issues (§4.1.2, §4.1.3). We do not use packet loss as a primary event-detection signal. This choice is consistent with measurements from major content providers, which often characterize Internet performance through latency and throughput rather than treating loss as a primary detection signal [88]. At the granularity of HERMES's user groups, loss is difficult to interpret on its own: it may arise from local wireless conditions, access-link behavior, or short-lived queueing, rather than from a persistent degradation along the network path. Persistent loss large enough to affect users should reduce throughput through retransmissions, and congestion severe enough to cause loss may also appear as elevated latency through queueing. We therefore detect events using throughput and latency, which capture the main user-visible consequences of sustained degradation, and use packet loss only as supporting diagnostic information.

4.1.1 User groups. We define user groups as pairs of ⟨AS, metro⟩, following common practices in network operations [11, 51]). This grouping allows us to attribute performance deviations to network or upstream problems rather than individual users' equipment. To reduce the risk of capturing anomalies unrelated to broader network conditions, we only consider a User Group if it includes measurements from at least 5 distinct IP addresses conducting collectively a minimum of 25 speed tests in the week prior to the event. Additionally, if a single IP address contributes more than 20% of the total speed tests, we downsample its excess. We perform a sensitivity analysis in Appendix B.4 and show in Section 5.3 that HERMES covers a large number of networks and metros worldwide despite this requirement. To ensure accurate grouping, we validate client locations using multiple geolocation sources and exclude tests with conflicting location data (details in Appendix A.1).

4.1.2 Latency. We detect latency anomalies using end-to-end RTT, as computed by the server's transport protocol (i.e., BBRv1 [12]). We set the baseline latencies to be the mean and median latencies across all speed tests from a user group in a week, similar to Microsoft's BlameIt [47]. To address temporal variability in speed tests, we aggregate data within hourly windows, using the median latency of each hour. This approach prevents any single window from disproportionately affecting the baseline latency and we remove clear outliers where the latency is above 5 s ($<$ 0.001% of our measurements).

To detect daily latency anomalies, we apply Welch's t-test and the Mann-Whitney U test to detect distributional changes. Welch's t-test identifies shifts in mean latency, while the Mann-Whitney U test assesses whether the daily distribution of latencies differs from the baseline by determining if a random observation from the baseline is smaller than one from the day of interest. Both yield a p-value indicating whether the observed and baseline latencies likely come from the same distribution. A low p-value signals a statistically significant latency increase, reducing the risk of misclassifying noise as an anomaly.

However, while these tests effectively detect statistically significant changes between distributions, they do not measure the *magnitude* of those changes. Even a minor increase of 1 ms consistently observed across the monitoring window could trigger an anomaly alert. To prevent alerts caused by small fluctuations, we introduce a sensitivity threshold, ϵ, which filters out minor deviations and highlights only substantial changes. We also require that more than 80% of latency measurements during the day of interest exceed the baseline median latency plus the sensitivity threshold. The 80% threshold ensures that anomalies reflect widespread changes within the User Group and is consistent with the value used by BlameIt [47]. A user group is experiencing a latency anomaly only when (i) the statistical tests yield a p-value below 0.05, and (ii) at least 80% of measurements exceed the baseline median plus ϵ.

4.1.3 Throughput. Detecting throughput anomalies is harder than detecting latency anomalies because throughput is both noisy and its distribution is composition-dependent. Users in the same ⟨AS, metro⟩ group may have different subscription plans [75], access technologies, devices, and Wi-Fi conditions [91]. Throughput distributions are thus often high-variance and multimodal: in our data, approximately 82% of user groups have higher normalized variance for throughput than for latency over the same baseline window, where variance is normalized by the squared mean to make the comparison unitless. The set of users running tests also changes from day to day, so a change in the mean or median may simply reflect a different mix of testers rather than a network degradation. At the same time, a real degradation may affect only one mode of the distribution or shift probability mass into a lower-throughput regime without changing the aggregate in a stable or interpretable way. HERMES therefore treats the throughput distribution itself, rather than any single aggregate statistic, as the monitored object. It compares the previous week's baseline distribution with the current-day distribution using the one-dimensional Wasserstein distance, which captures both how much probability mass moves and how far it moves across the throughput scale.

Permutation test. In a controlled measurement system, we might collect additional probes from the same user group to estimate this variability directly. HERMES instead observes only the user-triggered tests that happen to be run on each day, so the detector must quantify uncertainty from this finite set of measurements. We avoid a parametric model for throughput because throughput depends on subscription tiers, devices, home-network conditions, and user-triggered sampling. The test relies on a simple null hypothesis: if there is no distributional change, then the baseline and current-day measurements are samples from the same underlying throughput

distribution. In that case, the labels *baseline* and *current day* carry no statistical information; any measurement could have appeared in either sample. The labels are therefore exchangeable. HERMES pools the two samples, repeatedly assigns the pooled measurements at random into baseline and current-day groups of the original sizes, and recomputes the Wasserstein distance for each relabeling. The resulting empirical null distribution across many relabelings captures the distances expected from sampling variation alone, and the p-value is the fraction of permuted distances at least as large as the observed one. This test, called a *permutation test*, identifies distributional change, not necessarily network degradation. HERMES therefore uses it only as a screening step, then applies two additional requirements that guarantee the changes are degradations. First, as in the latency detector, we require a significant shift in central tendency using the Mann-Whitney U test. Second, we require the shift to be in the degraded direction: at least 80% of current-day measurements must fall below the baseline median minus the throughput sensitivity threshold δ. These requirements make the detector conservative: HERMES flags a throughput event only when the current-day distribution changes significantly, the median also shifts, and most measurements move toward lower throughput.

4.1.4 Parameter Sensitivity. We evaluate the impact of detection parameters in Appendix B.4. Varying these parameters reveals predictable trade-offs between sensitivity, coverage, and statistical confidence. Larger values of ϵ reduce the number of detected events by concentrating detections on higher-amplitude degradations, while stricter p-value thresholds increase confidence at the cost of biasing detection toward user groups with denser measurements. Increasing the required fraction of measurements that must exceed the baseline beyond 80% reduces detections by restricting anomalies to degradations that are consistently observed throughout the day. We report results using conservative defaults selected through manual inspection across diverse networks. HERMES also exposes all underlying detection quantities (e.g., p-values, deviations from baseline), letting operators tune thresholds based on their goals. An interesting open question is to invert this perspective and ask, given the empirical distribution of performance metrics for a ⟨AS, metro, site⟩ user group, how many measurements are sufficient to reliably detect degradations at a desired confidence level? Such an adaptive notion of "sufficient coverage" could further improve our coverage and sensitivity but exploring it rigorously is beyond the scope of this paper.

4.2 Building a Topology

To localize performance problems, we construct a topology that captures traffic flows between users and M-Lab servers. Each speed test is paired with a forward traceroute from the server to the client, and for 25% of tests[2], the Reverse Traceroute Sidecar [48, 81, 97] collects a reverse traceroute from the client back to the server, providing a bidirectional view of each path. As shown in Appendix B.6, 52–66% of reverse traceroutes are trustworthy, consistent with prior work [97]. Aggregating all measurements over the eight-day monitoring window (baseline week + event day) yields an IP-level graph of observed hops, which we enrich with AS, organization, and geographic annotations (Appendix A.2).

We represent each node as a ⟨AS, metro⟩ pair. This level of abstraction strikes a balance between fidelity and scalability: it is fine-grained enough to capture operator-relevant events (e.g., a congested peering link, a metro-level outage, or an AS-wide disruption) while coarse enough to avoid the instability and ambiguity of router-level views, where frequent interface changes, aliasing, and missing hops make attribution unreliable. A link is defined as a directed edge between two such nodes, capturing traffic flow between an AS-metro pair and its next hop. This representation is both interpretable for operators and scalable to Internet-wide analysis. The resulting graph serves as the substrate for our event localization algorithms (§4.3).

4.3 Pinpointing Responsible Entities

Pinpointing the sources of events from end-to-end measurements—known as network tomography—is a long-standing challenge. The goal is to infer which network entities (nodes or links) are responsible for observed problems using only end-to-end measurements. This problem is often under-specified: there are fewer independent measurements than unknowns, so multiple combinations of failures may explain the same observations. This ambiguity is well documented in prior work [28, 33, 55]. Events can also occur at multiple levels of granularity—from peering links to fibers, facilities, metro, or entire ASes [96]. Mapping across these levels adds complexity because failures at one granularity can mask or cascade into others.

Traditional network tomography methods rely on assumptions that do not hold at Internet scale. Many require dense and controlled measurement coverage [33] or dedicated probing infrastructure [23], while others use linear inference models that are sensitive to noise and costly to run at Internet scale [38]. Prior studies have also shown that general-purpose inference algorithms are computationally expensive [15, 16, 26, 66], making them infeasible to run across the hundreds of thousands of ⟨AS, metro⟩ pairs in our topology. These constraints motivate our solution.

Localization outputs. Algorithm 1 summarizes how HERMES turns detected events into localization outputs. For each detected event, HERMES returns one of three output states: (i) *localized*, when one or more entities pass our attribution thresholds and explain the event; (ii) *ambiguous*, when multiple entities remain plausible because existing measurements cannot distinguish them; and (iii) *unresolved*, when the available paths do not provide enough evidence to identify any potentially responsible entity. In all cases, HERMES reports the candidate entities, the anomalous paths that traverse them, and the temporal or correlation scores used to classify the event.

When a user group's performance deviates significantly from baseline as defined in Section 4.1, we refer to its forward and reverse paths as anomalous paths. HERMES localizes disruptions at multiple levels of granularity using two complementary techniques: *Temporal Tomography* highlights entities whose involvement in anomalous paths changes sharply from the baseline period to the event day, surfacing entities that become newly involved (or avoided) during the event (§4.3.1). *Correlation tomography* isolates entities that appear disproportionately often across anomalous paths (§4.3.2). This dual approach is necessary because not all anomalies stem

[2]Since completing this evaluation, we have doubled the operational reverse-traceroute sampling rate to 50%, improving bidirectional visibility for future event attribution.

Algorithm 1: HERMES event localization.

Input: Events $\mathcal{E}$; baseline paths P_b; event-day paths P_e; attribution threshold η; minimum path count $p_{\min}$
Output: Attribution state $s_e \in \{\text{LOCALIZED, AMBIGUOUS, UNRESOLVED}\}$ for each event $e \in \mathcal{E}$

// Build topology from bidirectional paths
$G \leftarrow$ AS–metro graph from forward and reverse paths in $P_b \cup P_e$;
foreach *event* $e \in \mathcal{E}$ **do**
  $U_e \leftarrow$ anomalous source–destination pairs for event e; $P_e(U_e) \leftarrow$ event-day paths for pairs in U_e; $T_e \leftarrow \emptyset$; $C_e \leftarrow \emptyset$;
  // Temporal tomography: entities whose path involvement changes during the event (§4.3.1)
  foreach *link* $\ell \in G$ *observed in at least* $p_{\min}$ *paths across* $P_b \cup P_e(U_e)$ **do**
    $p_b(\ell) \leftarrow$ fraction of baseline paths in P_b that traverse ℓ;
    $p_e(\ell) \leftarrow$ fraction of anomalous event-day paths in $P_e(U_e)$ that traverse ℓ;
    $\Delta_\ell \leftarrow p_e(\ell) - p_b(\ell)$;
    if $|\Delta_\ell| > \eta$ **then**
      $T_e \leftarrow T_e \cup \{\ell\}$
  // Correlation tomography: entities disproportionately shared by anomalous pairs (§4.3.2)
  $\mathcal{X}_e \leftarrow$ candidate entities at link, ⟨AS, metro⟩, AS, metro, and IXP granularities, observed $> p_{\min}$ event-day paths; $R \leftarrow U_e$;
  while $R \neq \emptyset$ **do**
    foreach $x \in \mathcal{X}_e$ **do**
      $\rho_R(x) \leftarrow$ fraction of x's observed paths that are anomalous, over pairs not yet explained ; // anomaly rate of x
    $x^* \leftarrow$ entity with $\rho_R(x^*) \geq \eta$ whose paths cover the most pairs in R;
    if *no such entity exists* **then break**;
    if x^* *is coarse and its anomalies concentrate in a single sub-entity* y *with* $\rho_R(y) \geq \eta$ **then**
      $x^* \leftarrow y$
    $C_e \leftarrow C_e \cup \{x^*\}$;
    $R \leftarrow R \setminus \{$pairs covered by $x^*\}$;
  // Assign attribution state
  $A_e \leftarrow T_e \cup C_e$;
  if $A_e = \emptyset$ **then**
    $s_e \leftarrow$ UNRESOLVED;
  else if *candidates in* A_e *are always co-traversed on the observed paths* **then**
    $s_e \leftarrow$ AMBIGUOUS; Schedule additional probes to use paths that separate the co-traversed candidates ; // Probing to disambiguate (§4.4)
  else
    $s_e \leftarrow$ LOCALIZED;

from rerouting—congestion may degrade performance along stable paths, while load-balancing changes may shift routes without causing degradation. Our algorithm draws inspiration from 007 [4], which operates in highly structured data center networks by allowing each anomalous path to "vote" for the components it traverses and attributing failures to components receiving the most votes. In Internet-wide measurements, however, such voting does not work well: paths are asymmetric and sparsely observed, visibility is uneven, and large transit networks naturally accumulate votes even when they are not the source of an event. Instead, HERMES combines temporal changes and cross-path correlation over anomalous paths to localize responsible entities under sparse visibility. Temporal and correlation tomography are run in parallel and capture different evidence. Temporal tomography identifies entities that appear much more or much less often on anomalous paths during the event day than during the baseline period, making it well suited to detecting rerouting events. Correlation tomography identifies entities disproportionately shared by anomalous paths, making it well suited to detecting congestion. The two procedures need not agree, and they may identify different parts of the same event (e.g., a failed interconnection that triggers a long degraded fallback path).

4.3.1 Temporal Tomography. To identify links most likely responsible for an event at time t_{event}, we compare each link's anomaly rate during the event against its baseline. For each link ℓ, we compute the fraction of observed paths containing ℓ before ($f_{t_{\text{before}}}(\ell)$) and during the event ($f_{t_{\text{event}}}(\ell)$) that are anomalous (i.e., paths whose associated tests exhibit unusually high latency or low throughput as defined in Section 4.1). We define the link's impact as $\Delta_\ell = f_{t_{\text{event}}}(\ell) - f_{t_{\text{before}}}(\ell)$. *Before* refers to the 7-day baseline period used to establish anomaly thresholds; *during* refers to the single day labeled as anomalous. For multi-day events, we analyze each anomalous day independently. A positive Δ_ℓ indicates a sharp rise in anomalous traffic traversing ℓ, while a negative value may indicate that ℓ disappeared or was avoided, potentially forcing traffic onto less optimal paths. Links that disappear entirely are assigned $f_{t_{\text{event}}}(\ell) = 0$. To reduce noise, we only analyze links observed in at least 10 paths during the event window, since with fewer than 10 samples the confidence interval on anomaly rates becomes too wide, making reliable attribution impossible. For example, consider a link $A \rightarrow B$ that is present in 10% of the paths taken by user groups both before and during an event. Before the event, only 2% of these paths were anomalous; during the event, 80% were anomalous, yielding $\Delta_\ell = 0.78$. This large shift suggests that $A \rightarrow B$ became impaired or began contributing to downstream degradation. While Temporal Tomography highlights links impacted by an event, it cannot always identify root causes: unobserved changes elsewhere in the network can trigger rerouting and anomalies [46]. Nonetheless, Δ_ℓ provides a signal of which links were directly affected.

4.3.2 Correlation Tomography. Not all performance issues stem from route changes. Congestion can degrade performance without altering the path, and a routine load-balancing update may trigger route changes without affecting performance. To capture these scenarios, we localize the network entity most likely responsible for an event from the forward and reverse traceroutes of the user groups experiencing the event on the target day. Rather than fix a single granularity, we build a topology of candidate entities at every level it exposes—individual links, ⟨AS, metro⟩ nodes, ASes, metros, and IXPs—and let the localization choose the granularity the evidence supports.

We score each candidate entity with two complementary quantities. The first is its *coverage*: the share of the remaining anomalous source–destination pairs whose paths traverse it, indicating how much of the event it could explain. The second is its *anomaly rate*: the fraction of the entity's own observed paths that are anomalous, i.e. the accuracy of blaming it. Coverage alone overrepresents large transit providers, which appear on many paths; the anomaly rate alone is unreliable for sparsely observed entities that may look anomalous by chance. Used together, coverage identifies entities that explain many anomalies, while the anomaly rate prevents attributing an event to an entity whose traffic is largely healthy.

We identify culprits with an iterative procedure. At each step we select, among entities whose anomaly rate exceeds a sensitivity threshold η, the one that explains the largest number of still-unexplained anomalous pairs. The pairs it explains are removed, and both coverage and the anomaly rate are recomputed on the *remaining* anomalies before the next step. This re-evaluation on the residual is essential: an entity that appears anomalous only because it carries one badly affected sub-region ceases to qualify once that sub-region has been explained, preventing a localized fault from being misread as a broad one. We repeat until no entity remains sufficiently anomalous, following the intuition that a single event is unlikely to originate from many unrelated parts of the network at once [4, 47]; each anomalous pair is thus attributed to a single most-responsible entity.

Because candidates span granularities, each selection also fixes a granularity. We report the *finest* entity consistent with the evidence and coarsen—from a link to its ⟨AS, metro⟩ node, or from nodes up to an AS, metro, or IXP—only when the anomaly is spread across several of its sub-entities rather than driven by a single one. For example, if the affected paths through a large transit AS concentrate in a few of its metros, we attribute the event to those ⟨AS, metro⟩ nodes; we implicate the entire AS only when its anomalies are spread across its footprint. This approach yields the most specific accurate explanation, avoiding over-generalizing a localized fault to a whole AS or metro while still recognizing genuinely wide failures.

Correlation tomography should be interpreted as inference over the paths HERMES observes and the entities those paths traverse. If two candidate entities always appear on exactly the same observed paths, then they are observationally indistinguishable: every anomalous path implicating one also implicates the other. In this case, no algorithm using only these measurements can determine which entity is the true source, so the correct output is an ambiguity set rather than a forced attribution. On the opposite, when the observed paths separate candidates—that is, true sources have anomalous paths that are not equally explained by non-sources—the greedy procedure recovers the most likely explanation. We formalize this in Appendix A.3.

4.4 Probing to Disambiguate

Our tomography algorithms often leave ambiguity sets—groups of adjacent links or nodes that appear equally plausible because available traceroutes cannot distinguish between them [33]. Resolving these ambiguities requires additional active measurements that are likely to traverse one candidate link but not the others. However, probing capacity is inherently limited: NDT tests require user cooperation, and large-scale active campaigns risk overloading measurement servers or ISPs. Given the Internet's size, naively probing every path is infeasible, making careful probe selection essential.

We therefore use a post-hoc measurement planning step that directs forward and reverse traceroutes from M-Lab sites where they have the most diagnostic value. Because on-demand NDT tests would overload client connections, we rely instead on forward and reverse traceroutes from additional M-Lab sites. Latency from these traceroutes serves as a proxy for a speed test latency, providing additional path-level constraints that help rule out incorrect explanations. Importantly, this probing is not intended to retroactively diagnose the triggering event. Rather, it reduces ambiguity so that future recurrences of similar events can be attributed more precisely. To guide this process, we build on metAScritic [87], a framework for uncovering hidden AS links within metros. One of its core components is a probability matrix $\mathbb{P}$, where each entry represents the highest estimated likelihood that some vantage point-destination path will traverse a given candidate peering link ℓ_{ij} (along with metadata indicating which measurement is expected to uncover the link with that probability). In our setting, we repurpose $\mathbb{P}$ as a probe-planning tool: given an ambiguity set S, we select (M-Lab site, user group) pairs whose traceroutes are predicted to have a high probability of crossing one candidate link while simultaneously having low probability of crossing the others. By greedily issuing those traceroutes, we maximize the diagnostic value of each measurement: a single observed path can confirm that a specific link is responsible for the anomaly while simultaneously ruling out others, rapidly shrinking the ambiguity set. Because probing resources are finite, we prioritize ambiguity sets by their impact—first the number of user groups affected and second the severity of degradation, measured as deviation from baseline median performance. We run active probing in daily cycles. When a path is admitted to the probe set to resolve an ambiguity, it joins a persistent probe roster and is re-probed once every 30 minutes until it is explicitly evicted. If a user group (and its associated paths) shows no anomalies for 10 consecutive days, we remove it from the roster to free capacity for emerging issues. Newly collected probes are folded back into the dataset and baselines, shrinking future ambiguity sets and steadily improving our topological coverage.

5 Evaluation

We evaluate whether HERMES can turn sparse, user-triggered speed tests into reliable Internet performance observability. In particular, we answer three questions: (i) Does the detector avoid spurious alarms that would disappear under richer telemetry? (ii) Does HERMES detect real user-facing degradations reported by operators and users? (iii) When HERMES detects an event, does it provide an accurate attribution for where the degradation occurs?

Section 5.1 evaluates event precision by comparing HERMES against a reimplementation of BlameIt [47], a cloud-scale monitoring reference built from real user traffic. This comparison asks whether events detected from sparse public data are corroborated by a richer cloud view. We also evaluate attribution accuracy by asking whether HERMES's inferred source is consistent with the coarse path segment identified by the reference system. Section 5.2

evaluates event recall against documented disruptions reported by operators, mailing lists, and users. Section 5.3 evaluates where HERMES has enough measurement density and path visibility to support Internet-scale monitoring, including both the number of speed tests within user groups and the representativeness of paths to the M-Lab servers compared to popular destinations. Finally, Section 5.4 and Section 5.5 evaluate the mechanisms that make attribution more accurate: bidirectional path visibility and targeted probing to reduce ambiguity. Validating these questions is challenging because there is no complete ground truth for Internet performance degradations or their causes, which is precisely the gap HERMES aims to address. We therefore compare HERMES against multiple independent datasets, using agreement as evidence of correctness while recognizing that no single dataset provides complete visibility. Because attribution cannot always be definitively verified, HERMES emphasizes transparency: every flagged event is accompanied by its supporting measurements and reasoning through a public dashboard [84] (described in detail in Appendix C.1), enabling operators to inspect results.

Key results. Despite relying only on sparse, public speed tests, HERMES detects events that are corroborated by connections of a large cloud provider in 91.4% of cases, establishing a lower bound on event-level precision (§5.1). Among events visible to both systems, HERMES's attribution is consistent with BlameIt [47] reference attribution in 94.5% of cases. HERMES also identifies between 64.7% and 85.1% of performance problems reported in ISP outage pages, operator-verified tickets, mailing lists, and user reports (§5.2). Its measurements span ASes hosting over 95% of the global Internet population and include sufficient data in many large metro areas to support reliable detection (§5.3). Bidirectional paths are central to attribution: 50.5% of implicated links require reverse-path visibility, and reverse-path rerouting explains a larger fraction of latency degradations than forward paths (§5.4). Targeted measurements reduce source ambiguity by 47% on average and fully resolve 31% of ambiguous cases (§5.5).

5.1 Comparing to a Cloud's Monitoring

To evaluate HERMES's precision, we use a reimplementation of BlameIt, a peer-reviewed system for localizing faults from server-side passive measurements of client connections to Microsoft Azure (supplemented for some incidents with active traceroutes) [47]. Microsoft's original BlameIt deployment is not available to us, so we reimplement its core localization methodology using passive latency measurements of client connections to frontend servers delivering services provided by AnonCDN, a network with similar scale.

Our reimplementation follows BlameIt's high-level structure. It groups measurements by client prefix, frontend site and short time window. Using the passive measurements, it then classifies each group as good or bad using latency thresholds and attributes degradations to one of three coarse path segments: the client side, the middle of the Internet path, or AnonCDN's WAN. When the BlameIt-style reference system attributes an event to the middle segment, we treat this as comparable to HERMES identifying an intermediate network entity along the path. For some events, BlameIt refined the localization using forward traceroutes from the cloud, but we did not implement this component.

Results: We cannot disclose exact telemetry counts from AnonCDN, but the reference system uses orders of magnitude more passive measurements than HERMES observes user-triggered speed tests. We compare events at the same user-group granularity used by HERMES, after aggregating prefixes into user groups. A HERMES event is corroborated when BlameIt detects a degradation for the same client AS and metro during the same day. Overall, 91.4% of events detected by HERMES are also visible in the cloud-scale reference system, suggesting that HERMES raises few false alarms despite relying only on public measurements. This overlap is a lower bound on event-level precision: some events missed by the cloud-scale reference system may occur on paths visible to M-Lab but not on the path to AnonCDN. Among the events visible to both systems, HERMES agrees with the reference system's inferred path segment in 94.5% of cases. Combining event corroboration with this conditional source agreement, 86.5% of all HERMES events have both event-level and source-level agreement.

5.2 Detecting Documented Events

To evaluate recall, we compare events detected by HERMES and their inferred sources against two complementary datasets: operator-confirmed outages and publicly reported disruptions.[3] For operator-confirmed data, we use ISP outage pages (authoritative but sparse provider-reported incidents) and AnonCDN support tickets (operator-verified network issues affecting AnonCDN services). We crawl publicly-reported disruptions from the NANOG and Outages mailing lists [67, 78], long-standing forums where experts share and vet outage reports [6], and from Reddit, which provides large-scale but less reliable user-reported accounts of connectivity issues. By combining operator-curated data (high accuracy, low coverage) with public reports (broad coverage, variable reliability), we balance confidence and scale, ensuring validation captures both well-documented outages and less certain, user-impacting disruptions that traditional monitoring systems overlook. We provide full details of how these datasets were collected, filtered, and validated in Appendix B.1.

Table 2 summarizes HERMES's detection and classification performance across validation sources, listed in order of decreasing reliability. Among high-confidence sources, HERMES successfully detects 85.1% of ISP-reported outages, 78.1% of operator-validated AnonCDN tickets, and 73.8% of mailing list-reported disruptions. For end-users report on Reddit, we manually reviewed and confirmed a subset of reports that reflected genuine network events. Within this subset of confirmed events, HERMES detected 64.7%, and across all Reddit events, HERMES captured 57.2%. This represents 11× more issues detected than existing public observatories such as Cloudflare Radar and IODA (more details in Appendix B.2).

Operators' outage pages—authoritative confirmations: ISP-operated outage pages provide official acknowledgments of service disruptions [22, 39, 73, 90]. Because these pages typically display

[3]We attempted to replicate the methodology of prior work that identified outages using Google Trends [50]. However, changes to the Google Trends API have significantly complicated the process of automatically crawling Google Trends, making the methodology obsolete [32].

Table 2: HERMES performance across validation datasets. 'Event' evaluates detecting a disruption; 'Source' evaluates joint detection and correct attribution when available. Recall is not reported for BlameIt because AnonCDN will not share the data and Precision is not reported for human-curated datasets, which do not exhaustively report events.

		Event (%)		Source (%)	
Data	**Events**	**Recall**	**Prec.**	**Recall**	**Prec.**
AnonCDN's BlameIt	≈100Ks	–	91.4	–	86.5
ISP Outages	27	92.6	–	85.1	–
AnonCDN's Tickets	≈10s	84.1	–	78.1	–
Mailing Lists	42	73.8	–	–	–
Reddit (Confirmed)	213	64.7	–	–	–
Reddit	3207	57.2	–	–	–

only current events, we use the Internet Archive's Wayback Machine [41] to retrieve all historical snapshots dating back to August 2024. We manually identified and archived outage pages for 4 networks and cross-referenced them with HERMES's detections. The archived data primarily covers outages involving transit networks, making this dataset valuable to evaluating whether HERMES correctly attributes the source of an event.

Result: HERMES detected 25 of the 27 documented outages (92.6% event-level recall) and correctly identified the source for 23 of these outages (85.1% source-level recall).

Troubleshooting tickets from AnonCDN—operator-level validation: We compared the events detected by our system with those reported by AnonCDN's clients and investigated by in-house network operators. Our analysis focused on 10s of incidents where faults occurred outside AnonCDN's infrastructure or direct peering links, as we are primarily interested in events impacting public Internet paths. For these incidents, we cross-referenced the responsible network—identified by AnonCDN's network operators—with events detected by HERMES during the same time periods. Troubleshooting tickets capture a subset of real-world disruptions, as many issues go unreported. Clients may not always recognize them as a network problem, and ISPs may lack incentives to escalate every incident. Reflecting this disparity, the number of events detected by BlameIt is two orders of magnitude higher than the number of tickets received by AnonCDN.

Result: HERMES detected 84.1% of operator-validated events (event-level recall). In cases with sufficient measurements, HERMES incorrectly localized 4 events and misclassified 3 as non-anomalous. Most missed events were due to insufficient measurement density.

5.3 Coverage and Representativeness of Internet Users and Paths

HERMES's accuracy depends on whether clients in a user group isssue enough M-Lab speed tests to support detection, and whether paths to/from M-Lab traverse infrastructure relevant to ordinary user traffic. We evaluate this along three axes: (i) *geographic/user coverage*, which asks where HERMES has enough measurements to monitor user populations; (ii) *infrastructure visibility*, which asks how much Internet infrastructure appears on paths observed by

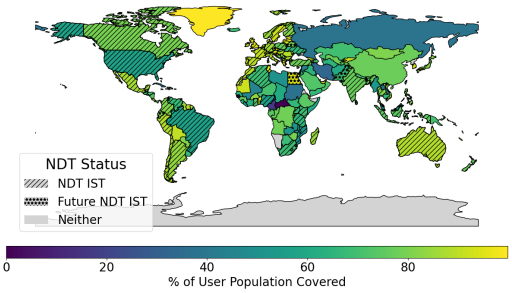


Figure 4: User population coverage by HERMES, with shading indicating the percentage of the population covered per country. Hatching denotes deployment status.

HERMES; and (iii) *path representativeness*, which asks whether paths to M-Lab overlap with paths toward popular Internet destinations.

Geographic and user coverage. Following the inclusion and downsampling criteria defined in Section 4.1.1, we consider an ⟨AS, metro⟩ user group covered when it has at least 25 tests from at least 5 distinct IP addresses over the baseline week for event detection. We evaluate sensitivity to these criteria in Appendix B.4.

We first ask whether HERMES covers networks that serve large user populations. No public dataset directly reports the number of users each AS serves in each metro, so we approximate this quantity using two complementary signals. First, we use APNIC's user-population dataset, which estimates the number of users served by each AS within each country [3, 85]. Second, we use IPInfo's user/hosting-prefix classification to identify prefixes that appear to host end users, and then map those prefixes to metros using IPInfo's geolocation dataset [44].

For each AS and country, we start with APNIC's estimate of the AS's user population. We then use IPInfo to identify the metros in that country where the AS has user-facing prefixes. Because we do not observe how the AS's users are distributed across those metros, we allocate users in proportion to metro population. This approach yields an estimated user population for each ⟨AS, metro⟩ pair. This estimate is approximate. It can be affected by IP geolocation error, incomplete AS footprints in IPInfo, and the assumption that users follow the metro population distribution. APNIC's estimates are also derived from Google Ads, so countries with limited Google presence, such as Russia, may be underrepresented [85, 101].

Figure 4 shows the resulting estimate of the fraction of Internet users in each country for which HERMES has sufficient measurements, based on the AS and metro of observed speed tests. Estimated coverage ranges from roughly 40% to 90% depending on the region, with the strongest coverage in Europe and North America and weaker coverage in parts of Africa. Some country-level estimates should be interpreted cautiously. In the United States, for example, coverage appears artificially low because large corporate and CDN networks are often geolocated to U.S. addresses; these prefixes inflate the denominator even though they do not map to residential users. In Brazil, APNIC reports a highly fragmented ecosystem of small access ISPs, many of which contribute only a small number of measurements, leading to under-coverage despite a large absolute user base [85]. At the metro level, coverage is strongest in high-population areas. In Appendix B.3.1, we show that 71.5% of the largest 10% of metros by population have at least

one ISP with enough measurements for continuous monitoring over the month.

Infrastructure visibility. User coverage does not by itself imply visibility into the network components that may explain performance degradations. Across the measurement period,HERMES observes 55.6% of all Points of Presences (PoPs) in iGDB, a database that maps networks to their infrastructure [2], indicating fairly broad coverage of the Internet despite relying on user-triggered tests. Appendix B.3.2 compares (at various granularities) the infrastructure traversed by measurements used by HERMES to measurements from CAIDA's Internet Topology Data Kit (ITDK) [10], which seeks to uncover as much infrastructure as possible.

Path representativeness. Finally, we ask whether paths to M-Lab resemble paths that carry high-volume Internet traffic. In Appendix B.3.3, we compare HERMES's observed paths against RIPE Atlas traceroutes toward popular destinations, including large CDNs and top websites from the Chrome UX Report [35]. HERMES overlaps with 80% of RIPE-observed ASes, 67% of metros, and 71% of ⟨AS, metro⟩ pairs on these paths. This does not mean that M-Lab paths represent all Internet traffic, nor that every destination would expose the same failures. It does show that paths to M-Lab traverse much of the same access, transit, and interconnection infrastructure used by popular Internet destinations. This overlap helps explain why, despite incomplete topology coverage, HERMES detects many documented user-facing performance events (§5.2).

5.4 Importance of Bidirectional Paths

To demonstrate why diagnosing network performance requires visibility into both the server-to-user (forward) and user-to-server (reverse) paths, we examine (i) how often performance changes correlate with shifts in the geographic or AS-level paths in either direction and (ii) how much of the observed latency can be accounted for by analyzing the forward path, the reverse path, or both together.

For (i), we evaluate whether latency and throughput degradations are associated with routing changes by comparing AS-level and geographic paths during anomalous measurements to those observed on the preceding day. If either the AS path or the sequence of metros changes in a given direction, we classify the event as involving a routing change in that dimension. This analysis only includes cases where we have both forward and reverse paths available (including baselines in the prior hour), but since the sampling is done fully randomly, this subset is likely representative of the broader dataset and the results we expect from HERMES as M-Lab increases the rate of reverse traceroutes. Table 3 shows that reverse path changes are more frequent than forward path changes, as 27.8% of events include a change in the reverse AS path and 26.6% include a change in the sequence of metros along the reverse path, compared to 19.8% and 24.1% in the forward path. By considering both paths together, 45.6% of the paths include a change in the AS path or the sequence of metros, highlighting the importance of bidirectional visibility in diagnosing network performance issues. At the same time, not all degradations coincide with routing changes, motivating the need for tomography methods that also capture congestion along stable paths (§4.3)

Table 3: Fraction of measurements with anomalous performance and routing changes across different dimensions.

Dimension	% of Paths	Dimension	% of Paths
Reverse AS Path	27.8	Only Reverse Paths	3.1
Forward AS Path	19.8	Only Forward Paths	1.5
Reverse Geo. Path	26.6	Both	25.5
Forward Geo. Path	24.1	At Least One	45.6

For (ii), in Figure 5, we estimate the percent of RTT that can be explained by geographic distance along the forward, reverse, and combined paths during events. We geolocate traceroute hops, compute path length, and convert distance to latency using the speed of light in fiber [92], ignoring erroneous geolocations and missing hops, yielding a lower bound on propagation delay. When analyzing a single direction, we assume the other follows the great-circle route. For example, in Section 3, we show that 70% of the observed latency could be explained by reverse path traffic being rerouted through Singapore, while only 17% was attributed to the forward path routing through Dallas. More broadly, the reverse path provides, on average, more insight than the forward path in explaining latency on paths during events. Finally, we show that source attribution degrades substantially when either forward or reverse paths are omitted: over 50% of event-causing links are identifiable only with bidirectional visibility (Appendix B.5). In Appendix C.3, we quantify routing asymmetry during anomalies by comparing forward and reverse path lengths, revealing that reverse paths are more circuitous in 72% of cases and at least twice as long in 10% of cases. Even when combining both directions, a substantial portion of latency remains unexplained by distance alone because of the limitations of traceroute probing (*e.g.*, layer-2 tunnels, unresponsive hops) and non-great-circle paths.

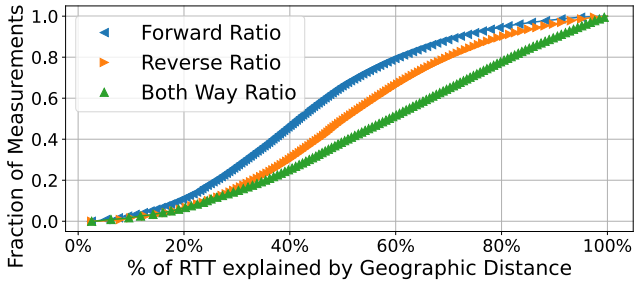


Figure 5: Distribution of the fraction of RTT explained by geographic distance for forward, reverse and both-way paths. In particular, the reverse path is more informative to explain the latency.

5.5 Targeted Measurements

Detected events may admit multiple plausible sources; HERMES reduces this ambiguity via targeted measurements (§4.4). To evaluate their effectiveness, we compare ambiguity set sizes before and after adding targeted probes. We allocate 10K reverse traceroutes per batch, issued every two hours (120K/day), and select probes using the algorithm in Section 4.4. Targeted traceroutes reduce the median ambiguity set size by 47%, from 2.4 to 1.7 edges. In 31% of cases, they fully resolve ambiguity, isolating a single edge as the likely source. We compare against a baseline that selects measurements uniformly at random; this baseline reduces ambiguity in 85%

fewer cases, highlighting the importance of measurement selection. While targeted probing substantially improves source identification, it does not resolve all cases: over a four-month period, we observe more than 15K events for which no additional measurement would reduce ambiguity, underscoring the limits imposed by server placement and coverage.

6 What can we see with HERMES?

Operational deployment: HERMES now runs continuously, producing updated event and attribution outputs each day. We are publishing these outputs as public BigQuery tables described in https://github.com/m-lab/hermes. Over a 5 month period, HERMES has processed approximately 4 billion speed tests, identifying 65K events across 16K cities in 166 countries, impacting clients in 9,710 ASes (see Appendix C.2 for daily aggregates). To ensure that HERMES's insights are accessible and actionable, we automated the system and developed a detailed dashboard (Appendix C.1).

Tracking anomalies across user groups and countries: We analyze the number of user groups that experience at least one anomaly during the 30-day monitoring window. Figure 6 offers an overview of the user groups with at least one anomaly across various countries. Throughput proves to be a critical metric for visibility, enabling the detection of significantly more anomalies than only latency. For instance, in the US, relying solely on latency would result in detecting approximately 2× fewer user groups experiencing events.

Mapping metro-level network disruptions during major events: The impact of real-world disruptions on Internet infrastructure is complex, extending beyond measures of network uptime. While prior works focus on outages [72, 89, 93], events like protests, cable cuts, or extreme weather can degrade the performance of PoPs and cables, leading to congestion or rerouting traffic onto longer, lower-performance paths without triggering outright outages on downstream users. HERMES provides a unique capability to quantify these types of disruptions, offering insights that complement traditional metrics of network liveliness.

Our primary case study examines the severe flooding in Andhra Pradesh, India, in September 2024 [94]. Figure 7 shows the change in the fraction of ASes experiencing anomalies in each metro before versus during the flooding. Metros near the most heavily flooded areas exhibit large increases (often exceeding 0.4), indicating widespread degradation across local networks. Notably, disruptions

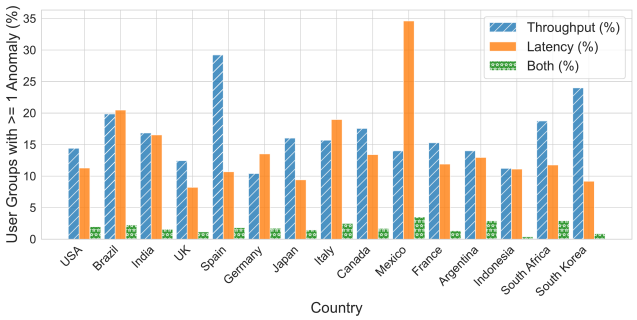


Figure 6: Percent of user groups with episodes of latency or throughput degradation in the 15 countries with the highest anomaly counts.

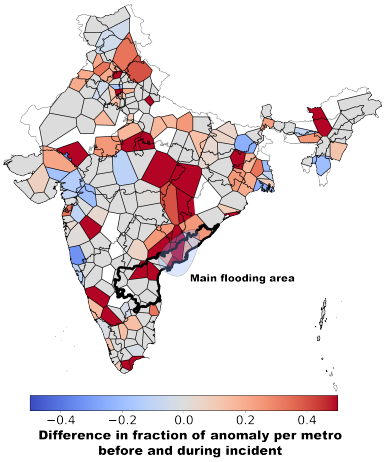


Figure 7: Difference in the fraction of user groups with detected anomalies across metro areas in India before and during the flooding.

extend beyond the directly impacted region, affecting neighboring metros as well, likely due to interdependencies in routing and shared infrastructure. Appendix C.5 presents analogous analyses for the Valencia flooding [5], Typhoon Shanshan [98], the Baltic Sea cable cut [7], and Hurricane Helene [31]. Appendix C.4 also demonstrates how HERMES infer interdomain congestion.

Diagnosing routing inefficiencies: HERMES's bidirectional path visibility enables diagnosis of routing inefficiencies that would be invisible from the forward path alone. By comparing forward and reverse path lengths during anomalies (Appendix C.3), we find that the reverse path is more circuitous in 72% of cases and at least twice as long as the forward path in 10% of cases. This asymmetry is consistent with prior work showing that optimizing the paths from the user to the server is more challenging [51, 102] and underscores that routing inefficiencies on the reverse path—which directly affect user-perceived latency—are missed entirely without bidirectional measurements.

Identifying persistently congested links: When paths remain unchanged before and during a degradation, HERMES uses correlation tomography to identify the network component most likely responsible. Table 7 in Appendix C.4 lists the interconnections most frequently implicated in congestion events. These often involve large transit providers (e.g., Tata–Bharti Airtel in Mumbai, Deutsche Telekom–Level 3 in Frankfurt, Vodafone–Level 3 in Milan). Unlike prior work on persistent congestion that required preselecting candidate links and running dedicated measurements to those links [25], HERMES repurposes already-running user-initiated speed tests and their accompanying paths, then launches targeted traceroutes only when additional path diversity is needed.

7 Conclusion

In this paper, we introduced HERMES, a system that leverages user-driven speed test data to monitor Internet performance and find the source of network issues. Using M-Lab data, HERMES accurately detects and localizes network events, as validated against operator-confirmed outages, a major CDN's monitoring system and public forums.

Acknowledgements

We thank our shepherd and the anonymous reviewers for their thoughtful feedback. We are grateful to the M-Lab team for their support and for maintaining the open measurement infrastructure that made this work possible. We also thank Anees Shaikh for carefully reading and commenting on the manuscript. This work was supported in part by NSF grant 2344761.

References

[1] Amazon Web Services. 2024. *Amazon CloudWatch Weather Map.* Amazon Web Services. https://aws.amazon.com/cloudwatch/ Accessed: 2024-12-03.

[2] Scott Anderson, Loqman Salamatian, Zachary S. Bischof, Alberto Dainotti, and Paul Barford. 2022. iGDB: Connecting the Physical and Logical Layers of the Internet. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 433–448. https://doi.org/10.1145/3517745.3561443

[3] APNIC. 2025. *Customers per AS Measurements – Visible ASNs: Customer Populations (Est.).* APNIC. https://stats.labs.apnic.net/aspop/ Accessed: 2025-09-17.

[4] Behnaz Arzani, Selim Ciraci, Luiz Chamon, Yibo Zhu, Hongqiang Harry Liu, Jitu Padhye, Boon Thau Loo, and Geoff Outhred. 2018. 007: Democratically Finding the Cause of Packet Drops. In *USENIX NSDI*. USENIX Association, Renton, WA, 419–435. https://www.usenix.org/conference/nsdi18/presentation/arzani

[5] Associated Press. 2024. *Massive Flooding in Valencia Due to Torrential Rains.* Associated Press. https://www.apnews.com/articles/valencia-2024-floods Accessed: 2024-10-29.

[6] Ritwik Banerjee, Abbas Razaghpanah, Luis Chiang, Akassh Mishra, Vyas Sekar, Yejin Choi, and Phillipa Gill. 2015. Internet Outages, the Eyewitness Accounts: Analysis of the Outages Mailing List. In *PAM*. Springer, Cham, Switzerland, 206–219. https://doi.org/10.1007/978-3-319-15509-8_16

[7] BBC News. 2024. *Severe Disruptions Due to Baltic Sea Cable Cuts.* BBC News. https://www.bbc.com/news/baltic-sea-cable-cuts-2024 Accessed: 2024-10-17.

[8] Karyn Benson, Alberto Dainotti, kc claffy, Alex C. Snoeren, and Michael Kallitsis. 2015. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 423–436. https://doi.org/10.1145/2815675.2815702

[9] David Blackwell and Joseph L. Hodges, Jr. 1957. Design for the Control of Selection Bias. *The Annals of Mathematical Statistics* 28, 2 (1957), 449–460.

[10] CAIDA. 2023. *The CAIDA Internet Topology Data Kit (ITDK).* CAIDA. https://www.caida.org/catalog/datasets/internet-topology-data-kit Accessed: 2024-01-30.

[11] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. 2015. Analyzing the Performance of an Anycast CDN. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 531–537. https://doi.org/10.1145/2815675.2815717

[12] Neal Cardwell, Ian Swett, and Joseph Beshay. 2024. *BBR Congestion Control.* Internet-Draft draft-ietf-ccwg-bbr-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-ccwg-bbr/01/ Work in Progress.

[13] Esteban Carisimo, Rashna Kumar, Caleb J. Wang, Santiago Klein, and Fabián E. Bustamante. 2024. Ten Years of the Venezuelan Crisis: An Internet Perspective. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 521–539. https://doi.org/10.1145/3651890.3672218

[14] Esteban Carisimo, Ricky K. P. Mok, David D. Clark, and kc claffy. 2022. Jitterbug: A New Framework for Jitter-Based Congestion Inference. In *PAM*. Springer, Cham, Switzerland, 155–179. https://doi.org/10.1007/978-3-030-98785-5_7

[15] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. 2004. Network Tomography: Recent Developments. *Statistical Science* 19, 3 (2004), 499–517. https://doi.org/10.1214/088342304000000422

[16] Yan Chen, David Bindel, Hanhee Song, and Randy H. Katz. 2004. An Algebraic Approach to Practical and Scalable Overlay Network Monitoring. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 55–66. https://doi.org/10.1145/1015467.1015475

[17] David R. Choffnes, Fabián E. Bustamante, and Zihui Ge. 2010. Crowdsourcing Service-Level Network Event Monitoring. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 387–398. https://doi.org/10.1145/2043164.1851228

[18] David D. Clark and Sara Wedeman. 2021. Measurement, Meaning and Purpose: Exploring the M-Lab NDT Dataset. In *TPRC*. TPRC, Washington, DC, USA, 44 pages. https://doi.org/10.2139/ssrn.3898339

[19] Cloudflare. 2024. *Cloudflare Radar.* Cloudflare. https://radar.cloudflare.com/ Accessed: 2024-12-04.

[20] Cloudflare. 2024. *Cloudflare Radar: United States Traffic on July 4th.* Cloudflare. https://radar.cloudflare.com/traffic/us?dateStart=2024-07-04&dateEnd=2024-07-06 Accessed: 2024-07-16.

[21] Cloudflare. 2025. *Cloudflare Speed Test.* Cloudflare. https://speed.cloudflare.com Accessed: 2025-04-30.

[22] Cogent Communications. 2025. *Cogent Network Status Page.* Cogent Communications. https://ecogent.cogentco.com/network-status Accessed: 2025-01-22.

[23] Ítalo Cunha, Renata Teixeira, Nick Feamster, and Christophe Diot. 2009. Measurement Methods for Fast and Accurate Blackhole Identification with Binary Tomography. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 254–266. https://doi.org/10.1145/1644893.1644924

[24] Omar Darwich, Hugo Rimlinger, Milo Dreyfus, Matthieu Gouel, and Kevin Vermeulen. 2023. Replication: Towards a Publicly Available Internet-Scale IP Geolocation Dataset. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3618257.3624801

[25] Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and kc claffy. 2018. Inferring Persistent Interdomain Congestion. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3230543.3230549

[26] Amogh Dhamdhere, Renata Teixeira, Constantine Dovrolis, and Christophe Diot. 2007. NetDiagnoser: Troubleshooting Network Unreachabilities Using End-to-End Probes and Routing Data. In *ACM CoNEXT*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/1364654.1364677

[27] Downdetector. 2024. *Downdetector: Real-Time Problem and Outage Monitoring.* Downdetector. https://downdetector.com Accessed: 2024-05-12.

[28] Nick Duffield. 2006. Network Tomography of Binary Network Performance Characteristics. *IEEE Transactions on Information Theory* 52, 12 (2006), 5373–5388.

[29] Mah-Rukh Fida, Andres F. Ocampo, and Ahmed Elmokashfi. 2021. Measuring and Localising Congestion in Mobile Broadband Networks. *IEEE Transactions on Network and Service Management* 19, 1 (2021), 366–380.

[30] Romain Fontugne, Cristel Pelsser, Emile Aben, and Randy Bush. 2017. Pinpointing Delay and Forwarding Anomalies Using Large-Scale Traceroute Measurements. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 15–28. https://doi.org/10.1145/3131365.3131384

[31] Fox Weather. 2024. *Hurricane Helene: Catastrophic Flooding and Widespread Destruction.* Fox Weather. https://www.foxweather.com/weather-news/top-weather-stories-2024 Accessed: 2024-10-09.

[32] General Mills. 2023. *Issue #561: API Changes Breaking PyTrends Usage.* GitHub. https://github.com/GeneralMills/pytrends/issues/561 Accessed: 2025-01-17.

[33] Denisa Ghita, Can Karakus, Katerina Argyraki, and Patrick Thiran. 2011. Shifting Network Tomography Toward a Practical Goal. In *ACM CoNEXT*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/2079296.2079320

[34] Petros Gigis, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett, and Georgios Smaragdakis. 2021. Seven Years in the Life of Hypergiants' Off-Nets. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 516–533. https://doi.org/10.1145/3452296.3472928

[35] Google Chrome Developers. 2026. *Chrome UX Report (CrUX) Documentation.* Google Chrome Developers. https://developer.chrome.com/docs/crux Accessed: 2026-02-05.

[36] 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, and Yiming Qiu. 2026. Skyline: A Cloud-Centric Internet Monitoring Engine. In *USENIX NSDI*. USENIX Association, Renton, WA, 685–699. https://www.usenix.org/conference/nsdi26/presentation/guo-shixian

[37] Vipul Harsh, Tong Meng, Kapil Agrawal, and Philip Brighten Godfrey. 2023. Flock: Accurate Network Fault Localization at Scale. *Proceedings of the ACM on Networking* 1, CoNEXT1, Article 3 (July 2023), 22 pages. https://doi.org/10.1145/3595289

[38] Yiyi Huang, Nick Feamster, and Renata Teixeira. 2008. Practical Issues with Using Network Tomography for Fault Diagnosis. *ACM SIGCOMM Computer Communication Review* 38, 5 (September 2008), 53–58. https://doi.org/10.1145/1452335.1452343

[39] Hurricane Electric. 2025. *Hurricane Electric Tunnel Broker Status Page.* Hurricane Electric. https://tunnelbroker.net/status.php Accessed: 2025-01-22.

[40] Geoff Huston. 2001. *Analyzing the Internet's BGP Behavior.* The Internet Protocol Journal. https://www.ece.ucf.edu/~yuksem/teaching/ip/reading/huston-bgp.pdf Accessed: 2025-01-17.

[41] Internet Archive. 2026. *Wayback Machine.* Internet Archive. https://web.archive.org/ Accessed: 2026-05-29.

[42] Internet Intelligence Research Lab, Georgia Institute of Technology. 2024. *Internet Outage Detection and Analysis (IODA).* Internet Intelligence Research Lab, Georgia Institute of Technology. https://ioda.inetintel.cc.gatech.edu/ Accessed: 2024-12-04.

[43] IODA. 2024. *IODA: ASN 174 (Cogent) Monitoring Data.* IODA. https://ioda.inetintel.cc.gatech.edu/asn/174?from=1720056703&until=1720229503 Accessed: 2024-07-16.

[44] IPInfo.io. 2025. *IP Geolocation API and Database.* IPInfo.io. https://ipinfo.io Accessed: 2025-01-24.

[45] IsItDownRightNow. 2024. *Is It Down Right Now? Website Down or Not?* IsItDownRightNow. https://www.isitdownrightnow.com Accessed: 2024-05-12.

[46] Umar Javed, Italo Cunha, David Choffnes, Ethan Katz-Bassett, Thomas Anderson, and Arvind Krishnamurthy. 2013. PoiRoot: Investigating the Root Cause of Interdomain Path Changes. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 183–194. https://doi.org/10.1145/2486001.2486036

[47] Yuchen Jin, Sundararajan Renganathan, Ganesh Ananthanarayanan, Junchen Jiang, Venkata N. Padmanabhan, Manuel Schroder, Matt Calder, and Arvind Krishnamurthy. 2019. Zooming In on Wide-Area Latencies to a Global Cloud Provider. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 104–116. https://doi.org/10.1145/3341302.3342073

[48] Ethan Katz-Bassett, Harsha V. Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter van Wesep, Thomas Anderson, and Arvind Krishnamurthy. 2010. Reverse Traceroute. In *USENIX NSDI*. USENIX Association, San Jose, CA, 219–234. https://www.usenix.org/conference/nsdi10-0/reverse-traceroute

[49] Ethan Katz-Bassett, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, David Wetherall, and Thomas Anderson. 2008. Studying Black Holes in the Internet with Hubble. In *USENIX NSDI*. USENIX Association, San Francisco, CA, 247–262. https://www.usenix.org/conference/nsdi-08/studying-black-holes-internet-hubble

[50] Ege Cem Kirci, Martin Vahlensieck, and Laurent Vanbever. 2022. "Is My Internet Down?": Sifting through User-Affecting Outages with Google Trends. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 290–297. https://doi.org/10.1145/3517745.3561428

[51] Thomas Koch, Shuyue Yu, Sharad Agarwal, Ethan Katz-Bassett, and Ryan Beckett. 2023. PAINTER: Ingress Traffic Engineering and Routing for Enterprise Cloud Networks. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 360–377. https://doi.org/10.1145/3603269.3604868

[52] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. 2009. Moving Beyond End-to-End Path Information to Optimize CDN Performance. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 190–201. https://doi.org/10.1145/1644893.1644917

[53] Matthew Luckie, Amogh Dhamdhere, David D. Clark, Bradley Huffaker, and kc claffy. 2014. Challenges in Inferring Internet Interdomain Congestion. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 15–22. https://doi.org/10.1145/2663716.2663741

[54] Matthew Luckie, Bradley Huffaker, Alexander Marder, Zachary Bischof, Marianne Fletcher, and kc claffy. 2021. Learning to Extract Geographic Information from Internet Router Hostnames. In *ACM CoNEXT*. Association for Computing Machinery, New York, NY, USA, 440–453. https://doi.org/10.1145/3485983.3494869

[55] Liang Ma, Ting He, Ananthram Swami, Don Towsley, Kin K. Leung, and Jessica Lowe. 2014. Node Failure Localization via Network Tomography. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 195–208. https://doi.org/10.1145/2663716.2663723

[56] Kyle MacMillan, Tarun Mangla, James Saxon, Nicole P. Marwell, and Nick Feamster. 2023. A Comparative Analysis of Ookla Speedtest and Measurement Labs Network Diagnostic Test (NDT7). *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 7, 1, Article 19 (March 2023), 26 pages. https://doi.org/10.1145/3579448

[57] Harsha Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. 2006. iPlane: An Information Plane for Distributed Services. In *USENIX OSDI*. USENIX Association, Seattle, WA, 367–380. https://www.usenix.org/conference/osdi-06/iplane-information-plane-distributed-services

[58] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, kc claffy, and Jonathan M. Smith. 2018. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 56–69. https://doi.org/10.1145/3278532.3278538

[59] Measurement Lab. 2014. *ISP Interconnection and Its Impact on Consumer Internet Performance.* Technical Report. Measurement Lab. https://www.measurementlab.net/publications/isp-interconnection-impact.pdf Accessed: 2025-01-24.

[60] Measurement Lab. 2024. *Network Diagnostic Tool (NDT).* Measurement Lab. https://www.measurementlab.net/tests/ndt/ Accessed: 2025-05-12.

[61] Measurement Lab. 2025. *MSAK (Measurement Swiss-Army Knife).* Measurement Lab. https://www.measurementlab.net/tests/msak/ Accessed: 2025-05-12.

[62] Measurement Lab. 2025. *Privacy Policy.* Measurement Lab. https://www.measurementlab.net/privacy/ Accessed: 2026-02-05.

[63] Measurement Lab. 2026. HERMES. https://github.com/m-lab/hermes/. GitHub repository.

[64] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. 2024. A Multifaceted Look at Starlink Performance. In *ACM WWW*. Association for Computing Machinery, New York, NY, USA, 2723–2734. https://doi.org/10.1145/3589334.3645328

[65] National Telecommunications and Information Administration. 2022. *Broadband Equity, Access, and Deployment (BEAD) Program.* National Telecommunications and Information Administration. https://broadbandusa.ntia.doc.gov/funding-programs/broadband-equity-access-and-deployment-bead-program Accessed: 2025-05-15.

[66] Hung Xuan Nguyen and Patrick Thiran. 2007. The Boolean Solution to the Congested IP Link Location Problem: Theory and Practice. In *IEEE INFOCOM*. IEEE, Piscataway, NJ, USA, 2117–2125. https://doi.org/10.1109/INFCOM.2007.245

[67] North American Network Operators' Group (NANOG). 2025. *NANOG Mailing List.* NANOG. https://mailman.nanog.org/mailman/listinfo/nanog Accessed: 2025-01-17.

[68] Lai Yi Ohlsen, Pavlos Sermpezis, and Melissa Newcomb. 2025. Poster: The Internet Quality Barometer Framework. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 1064–1065. https://doi.org/10.1145/3730567.3768593

[69] Ookla. 2025. *Ookla Open Data.* Ookla. https://www.ookla.com/open-data Accessed: 2025-01-27.

[70] Open Technology Fund. 2024. *OONI: Open Observatory of Network Interference.* Open Technology Fund. https://www.opentech.fund/projects-we-support/supported-projects/ooni-open-observatory-of-network-interference/ Accessed: 2024-12-09.

[71] OpenAI. 2024. *ChatGPT-4 API.* OpenAI. https://platform.openai.com/docs/ Accessed: 2024-12-04.

[72] Ramakrishna Padmanabhan, Aaron Schulman, Dave Levin, and Neil Spring. 2019. Residential Links under the Weather. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 145–158. https://doi.org/10.1145/3341302.3342084

[73] Path Network. 2025. *Path Network Status History Page.* Path Network. https://status.path.net/history Accessed: 2025-01-22.

[74] Udit Paul, Jiamo Liu, David Farias-llerenas, Vivek Adarsh, Arpit Gupta, and Elizabeth Belding. 2022. Characterizing Internet Access and Quality Inequities in California M-Lab Measurements. In *ACM COMPASS*. Association for Computing Machinery, New York, NY, USA, 257–265. https://doi.org/10.1145/3530190.3534813

[75] Udit Paul, Jiamo Liu, Mengyang Gu, Arpit Gupta, and Elizabeth Belding. 2022. The Importance of Contextualization of Crowdsourced Active Speed Test Measurements. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 274–289. https://doi.org/10.1145/3517745.3561441

[76] PeeringDB. 2026. *PeeringDB.* PeeringDB. https://www.peeringdb.com Accessed: 2026-06-24.

[77] Photon Shift Project. 2024. *Photon Reddit Data Download Tool.* Photon Shift Project. https://arctic-shift.photon-reddit.com/download-tool Accessed: 2025-01-24.

[78] Puck Nether Net. 2024. *Outages Mailing List.* Puck Nether Net. https://puck.nether.net/mailman/listinfo/outages Accessed: 2024-12-04.

[79] Lin Quan, John Heidemann, and Yuri Pradkin. 2013. Trinocular: Understanding Internet Reliability through Adaptive Probing. In *ACM SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 255–266. https://doi.org/10.1145/2486001.2486017

[80] Reddit user. 2024. *Is Cogent Down in Chicago?* Reddit. https://old.reddit.com/r/sysadmin/comments/1e41e7y/is_cogent_down_in_chicago/ Accessed: 2024-12-02.

[81] Reverse Traceroute team. 2025. *revtr-sidecar.* Reverse Traceroute team. https://github.com/NEU-SNS/revtr-sidecar Accessed: 2025-09-18.

[82] RIPE NCC. 2024. *RIPE IPmap: Infrastructure Geolocation.* RIPE NCC. https://labs.ripe.net/tools/ripe-ipmap/ripe-ipmap/ Accessed: 2024-12-09.

[83] RIPE Network Coordination Centre. 2026. *RIPE Atlas: A Global Network of Internet Measurement Probes.* RIPE NCC. https://atlas.ripe.net/ Accessed: 2026-02-05.

[84] Loqman Salamatian. 2026. HERMES Public Internet Incident Dashboard. https://hermes-dashboard.org/. Online dashboard.

[85] Loqman Salamatian, Calvin Ardi, Vasileios Giotsas, Matt Calder, Ethan Katz-Bassett, and Todd Arnold. 2024. What's in the Dataset? Unboxing the APNIC per AS User Population Dataset. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 165–182. https://doi.org/10.1145/3646547.3688411

[86] Loqman Salamatian and Phillipa Gill. 2025. *How M-Lab Determines User Location and Selects Servers.* Measurement Lab. https://www.measurementlab.net/blog/improving-m-lab-geolocation/ Accessed: 2025-05-27.

[87] Loqman Salamatian, Kevin Vermeulen, Italo Cunha, Vasilis Giotsas, and Ethan Katz-Bassett. 2024. metAScritic: Reframing AS-Level Topology Discovery as a Recommendation System. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 337–364. https://doi.org/10.1145/3646547.3688429

[88] Brandon Schlinker, Italo Cunha, Yi-Ching Chiu, Srikanth Sundaresan, and Ethan Katz-Bassett. 2019. Internet Performance from Facebook's Edge. In *ACM IMC*.

Association for Computing Machinery, New York, NY, USA, 179–194. https://doi.org/10.1145/3355369.3355567
[89] Aaron Schulman and Neil Spring. 2011. Pingin' in the Rain. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 19–28. https://doi.org/10.1145/2068816.2068819
[90] SFR. 2025. *SFR Network Status Page*. SFR. https://www.sfr.fr/media/export-arcep/siteshorsservices.csv Accessed: 2025-01-22.
[91] Ranya Sharma, Nick Feamster, and Marc Richardson. 2024. A Longitudinal Study of the Prevalence of WiFi Bottlenecks in Home Access Networks. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 44–50. https://doi.org/10.1145/3646547.3689007
[92] Sikich LLP. 2024. *Networking at the Speed of Light: Understanding Fiber Optics*. Sikich LLP. https://www.sikich.com/insight/networking-at-the-speed-of-light-understanding-fiber-optics/ Accessed: 2024-12-09.
[93] Xiao Song, Guillermo Baltra, and John Heidemann. 2023. Inferring Changes in Daily Human Activity from Internet Response. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 627–644. https://doi.org/10.1145/3618257.3624796
[94] Sphere India. 2024. *AP Flood Situation Report: Sitrep-1*. Sphere India. https://www.sphereindia.org.in/sites/default/files/2024-10/SI%20Sitrep-1__AP%20Flood%20Situation_02-09-2024.pdf Accessed: 2025-01-23.
[95] Srikanth Sundaresan, Xiaohong Deng, Yun Feng, Danny Lee, and Amogh Dhamdhere. 2017. Challenges in Inferring Internet Congestion Using Throughput Measurements. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 43–56. https://doi.org/10.1145/3131365.3131382
[96] William Sussman, Emily Marx, Venkat Arun, Akshay Narayan, Mohammad Alizadeh, Hari Balakrishnan, Aurojit Panda, and Scott Shenker. 2022. The Case for an Internet Primitive for Fault Localization. In *ACM HotNets*. Association for Computing Machinery, New York, NY, USA, 160–166. https://doi.org/10.1145/3563766.3564105
[97] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, David Choffnes, and Ethan Katz-Bassett. 2022. Internet-Scale Reverse Traceroute. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 694–715. https://doi.org/10.1145/3517745.3561422
[98] Wikipedia Contributors. 2024. *Typhoon Shanshan: Flooding and Widespread Damage*. Wikipedia. https://en.wikipedia.org/wiki/Typhoon_Shanshan_(2024) Accessed: 2024-08-28.
[99] WorldPop. 2025. *WorldPop Gridded Population Data*. WorldPop. https://www.worldpop.org/ Accessed: 2025-01-23.
[100] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. 2004. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *USENIX OSDI*. USENIX Association, San Francisco, CA, 167–182. https://www.usenix.org/conference/osdi-04/planetseer-internet-path-failure-monitoring-and-characterization-wide-area
[101] Zesen Zhang, Jiting Shen, and Ricky K. P. Mok. 2024. Empirical Characterization of Ookla's Speed Test Platform: Analyzing Server Deployment, Policy Impact, and User Coverage. In *IEEE CCWC*. IEEE, Piscataway, NJ, USA, 630–636. https://doi.org/10.1109/CCWC60891.2024.10427883
[102] Jiangchen Zhu, Kevin Vermeulen, Italo Cunha, Ethan Katz-Bassett, and Matt Calder. 2022. The Best of Both Worlds: High Availability CDN Routing without Compromising Control. In *ACM IMC*. Association for Computing Machinery, New York, NY, USA, 655–663. https://doi.org/10.1145/3517745.3561421

Appendix

Appendices are supporting material that has not been peer-reviewed.

A Methodology Details and Extensions

A.1 Cleaning User Geolocation Data

HERMES groups data by *metro*, so accurate geolocation is critical. M-Lab uses two independent systems to geolocate clients [86]. At test time, Google's internal geolocation service estimates the client's location from request metadata (e.g., HTML headers) to direct the user to the nearest M-Lab server. For the public BigQuery dataset, client IP addresses are annotated with MaxMind's GeoLite2 database. We treat agreement between them as a sign of correctness. We compute which server would be closest to the MaxMind-reported coordinates and compare it to the server chosen by Google's system. If the two disagree, we flag the test as potentially misgeolocated and exclude it from metro-level analysis. This approach leverages the independence of the two geolocation systems: measurements that pass this cross-check are far more likely to be accurately geolocated, while disagreements highlight cases where precise client location cannot be trusted. Applying these checks removes only a small fraction of tests (between 7.9% in South America and 18.4% in Africa), striking a balance between coverage and accuracy [86].

A.2 Adding Metadata to Traceroutes

IP to AS: We use the same method as prior work to map the IP addresses from traceroutes and reverse traceroutes to their ASes and IXPs [87, 97]. Tools like `bdrmapIT` [58] improve IP-to-AS mapping by leveraging external datasets (e.g., alias resolution datasets), but, at the ⟨AS, metro⟩ granularity used in this paper, these improvements add complexity but rarely affect results. Using CAIDA's ITDK dataset [10], we find that over 99.6% of interfaces map to the same ⟨AS, metro⟩ pair using both our approach and `bdrmapIT`, and fewer than 0.02% of paths differ for more than one hop.

IP to Facility: We map each hop to a possible facility in two steps. First, we map the hop IP to a metro area using the geolocation data. We then map that location to the set of possible facilities present in that metro according to iGDB [2].

IP to geolocation: To map each intermediary hop to its geolocation (latitude, longitude, city), we prioritize Hoiho [54] over RIPE IPMap [82], which are peer-reviewed techniques combining reverse DNS and latency information to derive a city level geolocation, over IPinfo [44], the geolocation database with the best performance, as recently demonstrated by prior work [24]. We remove IP addresses where the observed latency in the traceroute would imply a violation of the speed-of-light when accounting for the geographic path taken up to that hop. In particular, for each hop of a forward and reverse traceroute, we calculate the minimum feasible latency based on the sum of great-circle distances between consecutive hops. Because the reverse path is unknown for intermediate hops, we conservatively assume the shortest possible path (i.e., a direct great-circle distance) for the segment back. Latency is computed assuming light propagation in fiber ($\approx \frac{2}{3}$ of the speed of the light) unless the measurement is originating from Starlink where we assume light propagation in the vacuum. If the observed RTT for a hop is shorter than this theoretical minimum, we flag that hop as potentially misgeolocated and remove its geolocation.

A.3 Guarantees for Correlation Tomography

This section states the theoretical guarantee behind correlation tomography, conditioned on the accuracy of the event labels produced by Section 4.1 and the paths observed by HERMES. In particular, we characterize what can and cannot be identified from the observed relationship between anomalous measurements and the network entities they traverse.

Definition A.1 (Path-incidence explanation)**.** Fix one event day and one candidate granularity, such as links, ⟨AS, metro⟩ nodes, ASes, metros, or IXPs. Let $\mathcal{M}$ be the observed path measurements, $\mathcal{A} \subseteq \mathcal{M}$ the measurements labeled anomalous by Section 4.1, and $\mathcal{N} = \mathcal{M} \setminus \mathcal{A}$ the remaining measurements. Each candidate entity e has an incidence set

$$\mathcal{P}_e = \{m \in \mathcal{M} : e \text{ appears on the observed path of } m\}.$$

Write $\mathcal{A}_e = \mathcal{A} \cap \mathcal{P}_e$, and define the anomaly ratio

$$r(e) = \frac{|\mathcal{A}_e|}{|\mathcal{P}_e|},$$

for candidates with $|\mathcal{P}_e| > 0$. A set S explains the observed anomalies if

$$\mathcal{A} = \bigcup_{e \in S} \mathcal{A}_e.$$

Two entities e and e' are fully observationally indistinguishable if $\mathcal{P}_e = \mathcal{P}_{e'}$ on the measurements available to HERMES. This is a sufficient but not necessary condition for ambiguity: entities may remain difficult to distinguish even when their full incidence sets differ, for example if they are co-traversed on the anomalous measurements that drive the localization decision. We use the stronger notion here because it gives a model-free impossibility statement: when two entities have identical observed incidence, no algorithm using only these data can distinguish them without additional measurements or external information.

Lemma A.2 (Unavoidable ambiguity). *If two candidate entities are observationally indistinguishable, no tomography algorithm that uses only the path-incidence data $\{\mathcal{P}_e : e\}$ and anomaly labels $\mathcal{A}$ can determine which of the two entities is the true source.*

Proof. Suppose e and e' satisfy $\mathcal{P}_e = \mathcal{P}_{e'}$. For every observed measurement m, the statement "m traverses e" is identical to the statement "m traverses e'." Therefore the complete observation available to the algorithm is the same in a world where e is responsible and in a world where e' is responsible. Any algorithm must produce the same output on identical observations, so it cannot distinguish the two cases without additional measurements or external information. □

The lemma motivates the ambiguity sets reported by HERMES. Before stating the recovery result, we conceptually merge each set of observationally indistinguishable entities into a single equivalence class. Below, we refer to these equivalence classes as entities.

Theorem A.3 (Recovery under separability). *Consider the path-incidence model in Definition A.1, after merging observationally indistinguishable entities, and assume every candidate entity—in particular every true source—is observed on at least $p_{\min}$ paths, so that none is excluded by the support filter of Algorithm 1. Fix a threshold $\eta \in (0, 1]$. Let $S^\star$ be the true set of responsible entities, and assume $S^\star$ explains all observed anomalies. For any set $T \subseteq S^\star$ of already selected true sources, define the residual anomalous measurements*

$$\mathcal{R}_T = \mathcal{A} \setminus \bigcup_{e \in T} \mathcal{A}_e$$

and the residual anomaly rate[4]

$$r_T(e) = \frac{|\mathcal{R}_T \cap \mathcal{P}_e|}{|\mathcal{P}_e|}.$$

Assume the following separability condition holds: for every $T \subsetneq S^\star$, (i) every non-source $e \notin S^\star$ satisfies $r_T(e) < \eta$, and (ii) if $\mathcal{R}_T \neq \emptyset$, then some remaining true source $e \in S^\star \setminus T$ satisfies $r_T(e) \geq \eta$. Assume also that each true source has a witness anomalous measurement: for every $e \in S^\star$, there exists $m_e \in \mathcal{A}_e$ such that $m_e \notin \mathcal{P}_{e'}$ for all $e' \in S^\star \setminus \{e\}$. Then greedy correlation tomography terminates having selected exactly the entities in $S^\star$. Moreover, no strict subset of $S^\star$ explains all observed anomalies.

Proof. We prove the claim by induction over the greedy iterations. Initially $T = \emptyset$. Suppose that after some iterations the algorithm has selected only true sources, so the selected set is $T \subseteq S^\star$. If all anomalous measurements have been explained, the algorithm may stop. Otherwise $\mathcal{R}_T \neq \emptyset$, and the separability condition implies that at least one remaining true source has residual anomaly rate at least η. The same condition implies that every non-source has residual anomaly rate below η. Therefore any entity selected by the greedy rule—regardless of how ties among qualifying candidates are broken, and including an entity substituted by the granularity-refinement step, which must also satisfy $r_T(e) \geq \eta$—must be a true source. Adding it to T preserves the induction invariant. Moreover, the algorithm terminates: each selected entity satisfies $r_T(e) \geq \eta > 0$, so its selection explains at least one previously unexplained anomalous measurement, after which $r_T(e) = 0$; the residual shrinks strictly at every iteration, and the algorithm halts after at most $|S^\star|$ selections.

It remains to show that the algorithm cannot stop before selecting all true sources. Suppose $T \subsetneq S^\star$. Choose any $e \in S^\star \setminus T$. By the witness condition, there is an anomalous measurement m_e traversing e and no other true source. Since all selected entities are in T, m_e has not been explained, so $\mathcal{R}_T \neq \emptyset$. Separability then guarantees that some remaining true source still has residual anomaly rate at least η, so the stopping condition is not met. Thus the algorithm continues until all entities in $S^\star$ have been selected.

Finally, the witness condition implies minimality with respect to $S^\star$: removing any $e \in S^\star$ leaves its witness measurement m_e unexplained by the other true sources. Hence no strict subset of $S^\star$ explains all observed anomalies. □

Finite-sample interpretation. The goal of the separability condition is to state when the empirical scores used by correlation tomography are informative enough to distinguish a responsible entity from a co-traversed but non-responsible entity. In Algorithm 1, the algorithm maintains a residual set R of anomalous source–destination pairs that have not yet been explained, and scores each candidate entity x by its residual anomaly rate $\rho_R(x)$: the fraction of x's observed paths among not-yet-explained pairs that are anomalous—the empirical counterpart of r_T in Theorem A.3. The greedy step selects a candidate only if $\rho_R(x) \geq \eta$. Thus, η is an accuracy threshold on attribution: a selected entity's remaining observed traffic must be at least an η fraction anomalous. The minimum path-count threshold $p_{\min}$ prevents the algorithm from considering entities whose apparent anomaly rate is based on too few observed paths. The main finite-sample issue is co-traversal. A non-responsible entity can obtain a high anomaly rate if its observed paths largely coincide with those of a responsible entity. This is the same identifiability problem studied in prior fault-localization systems: if traffic crossing a healthy component overlaps too heavily with traffic crossing a faulty component, the healthy component

[4] Algorithm 1 computes this rate over the measurements of not-yet-explained pairs only; the separability condition and the proof are unchanged under either convention, applied to whichever rate the algorithm thresholds.

can receive too much "blame' ' for affected measurements [37]. Similar limits arise throughout network tomography and end-to-end fault localization [4, 23, 33, 38].

To make this intuition concrete, consider one residual iteration of the algorithm. Suppose paths that traverse a responsible entity are anomalous with probability p_1, while paths that do not traverse any responsible entity are anomalous with background probability $p_0 < p_1$. Let α denote the co-traversal overlap of a non-responsible candidate with the responsible entity: the fraction of the non-responsible candidate's observed paths that also traverse the responsible entity. The expected anomaly rate of the responsible entity is then p_1, while that of the non-responsible candidate is $\alpha p_1 + (1-\alpha)p_0$: when α is small, most of the candidate's observed paths avoid the responsible entity and provide counterevidence; when α is close to one, the candidate is almost always observed together with the responsible entity, so the two receive similar empirical scores. Under this simplified model, the expected separation between the score of a responsible entity and that of a co-traversed non-source is

$$\Delta = (1-\alpha)(p_1 - p_0).$$

This expression captures the two quantities that matter for correlation tomography. The term $p_1 - p_0$ is the strength of the anomaly signal: responsible paths must be more likely to appear anomalous than background paths. The term $1-\alpha$ is the amount of separating path evidence; the data must include paths that distinguish the responsible entity from nearby or co-traversed candidates. The bound below should therefore be read as an interpretation of when the empirical scores $\rho_R(x)$ preserve the ordering assumed by the greedy step. Suppose each candidate's score is computed from N observed paths whose anomaly indicators are independent—an idealization, since paths sharing infrastructure are correlated and successive iterations are coupled through the residual. If the expected score gap between true sources and non-sources is at least Δ, then it is enough for every empirical score to deviate from its expectation by less than $\Delta/2$. Hoeffding's inequality bounds the probability of a larger deviation for one candidate by $2\exp\left(-2N(\Delta/2)^2\right)$. Applying a union bound over m candidates gives

$$\mathbb{P}\big(\text{any score deviates by} \geq \Delta/2\big) \leq 2m\, e^{-N\Delta^2/2}.$$

Thus, to preserve the correct ordering with probability at least $1-\delta$, it suffices that

$$N \geq \frac{2}{\Delta^2} \ln\left(\frac{2m}{\delta}\right).$$

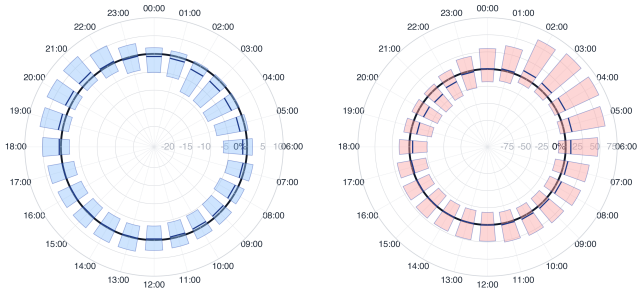
This finite-sample view explains the roles of the operational safeguards in Algorithm 1. The threshold $p_{\min}$ ensures that candidate scores are not computed from vanishingly small path support. The threshold η requires a candidate's observed traffic to be predominantly anomalous before it is selected, preventing attribution to entities whose traffic is largely healthy. The ambiguity state captures the case where candidates remain co-traversed on the observed paths, making their empirical scores difficult or impossible to separate. Finally, targeted probing helps only when it reduces α, as it adds paths that traverse one candidate without the other, thereby increasing the effective gap Δ. If additional measurements merely repeat the same co-traversed paths, they increase N but do not create separation; in the limit $\alpha \to 1$, the gap vanishes and the candidates are indistinguishable from the observed path-incidence data alone, the regime formalized by Lemma A.2.

A.4 A Proof-of-Concept for Sub-Daily Operation of HERMES

The core logic of HERMES does not fundamentally depend on daily aggregation. Daily aggregation treats within-day variation, including normal diurnal structure, as part of the recent day-level baseline distribution. This is appropriate for detecting sustained degradations that shift the day's measurements relative to recent comparable days, but it can be sensitive to large changes in the time-of-day composition of measurements. Identifying degradation on finer timescales therefore requires baselines that preserve normal diurnal variation in performance. We partition time into fixed time-of-day bins (e.g., one-hour or multi-hour windows) and compute a separate baseline for each bin using historical measurements from the same time-of-day. At finer granularity, the primary challenge is data availability: individual user groups may not generate enough measurements within a given bin to support statistically reliable detection. Accordingly, the minimum thresholds on the number of source IP addresses and measurements are enforced per bin, and events are detected by comparing current measurements against the baseline corresponding to that bin. We describe these adaptations below and discuss how reduced sample sizes affect variance and confidence at finer temporal resolutions.

A.4.1 Hourly Baselines. For each user group, we collect measurements from a rolling historical window and compute (i) a daily baseline using all measurements in the window as described in Section 4, and (ii) our bin baseline using only measurements that fall in the same local bin (e.g., all 8-9pm measurements if the bin size is 1 hour). This produces a bin-indexed reference distribution that captures diurnal structure (peak-hour slowdowns, off-peak recovery) that would be smeared out by daily aggregation. Figure 8 summarizes this comparison across all user groups using a radial boxplot representation, with one box per hour of day. Each box shows the distribution of hourly-daily baseline differences across user groups. Values above zero indicate that performance during that hour is worse than what would be implied by a daily baseline, while values below zero indicate better-than-daily performance. In particular, peak hours (18:00–22:00) exhibit consistently worse performance relative to daily baselines, while off-peak hours (1:00–10:00) tend to be faster.

A.4.2 Coverage Trade-off. Moving from daily aggregation to sub-daily bins introduces a fundamental coverage trade-off. Some user groups that are well covered at the day level lack sufficient measurements in finer-grained time bins to support statistically sound detection. Thus, a degradation that is detectable after pooling measurements over an entire day may not be detectable in any individual sub-daily bin: the signal can be split across bins, and each bin must independently satisfy the source-IP and measurement thresholds. Figure 9 quantifies this effect by showing, for each user group-site pair with sufficient coverage for daily analysis, how many sub-daily bins within a day also satisfy the minimum baseline requirements. At hourly granularity, coverage is sparse: with a 14-day lookback, fewer than 25% of user group-site pairs have



(a) RTT (b) Throughput

Figure 8: Comparison of hourly and daily performance baselines by hour of day. Radial boxplots show the distribution of percentage differences across user groups, highlighting the diurnal structure obscured by daily aggregation.

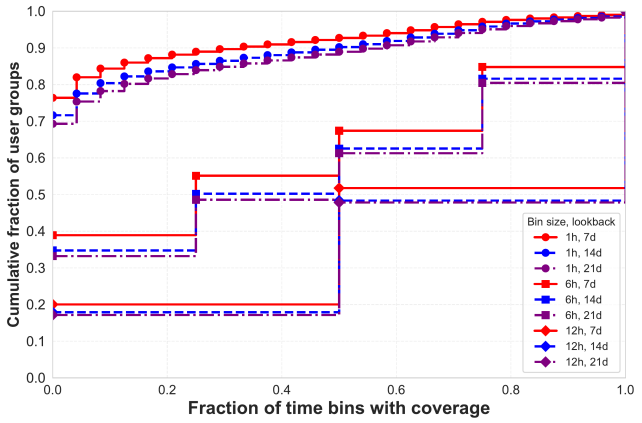


Figure 9: Sub-daily coverage under different lookback windows. Among user groups with enough measurements to run the daily version of HERMES, each CDF shows the fraction of sub-daily time bins with sufficient baseline measurements. Curves vary by sub-daily bin size (1h, 6h, or 12h) and baseline lookback window (7, 14, or 21 days). Longer lookback windows modestly increase sub-daily coverage by pooling more historical measurements, but with diminishing returns.

sufficient measurements in more than half of the 24 hours of a day, and fewer than 10% are covered in all hours. Extending the lookback window from 7 to 21 days increases hourly coverage, but only modestly, with clear diminishing returns.

A.4.3 Daily vs. Sub-Daily Event Detection. To better understand how sub-daily operation changes the events detected by HERMES, we compare daily detection against three bin-size instantiations of the pipeline: 1, 6, and 12 hours. For each $\langle AS, metro, site \rangle$ user group and day in December 2025, we classify outcomes into four categories: events detected by both daily and sub-daily analysis, events detected only at daily granularity, events detected only at sub-daily granularity, and days with no detected events at either resolution. Table 4 compares daily and sub-daily detection for bin sizes of 1, 6, and 12 hours. Intermediate bin sizes (6 hours) surface the largest fraction of sub-daily-only events, reflecting a balance between temporal resolution and statistical coverage. At the finest

Table 4: Comparison of daily and sub-daily event detection for different time bin sizes. Each entry reports the fraction of $\langle AS, metro, site, day \rangle$ pairs falling into each category.

Bin size	Both	Daily only	Sub-daily only	Neither
1 hour	1.2%	7.1%	6.8%	84.9%
6 hours	3.5%	4.8%	7.4%	84.3%
12 hours	4.7%	3.5%	5.5%	86.3%

granularity (1 hour), coverage constraints and reduced per-bin sample sizes can split otherwise detectable day-level degradations into underpowered fragments, reducing overlap with daily detections. Conversely, sub-daily detection can surface short-lived events that are diluted when measurements are pooled over an entire day. Across all bin sizes, events detected at sub-daily granularity typically span only a small number of bins of that day, confirming that most sub-daily-only events are short-lived.

Overall, these results show that daily detection strikes a balance between coverage, statistical power, and operational relevance (§5). Furthermore, most of the events discussed by users were detected using the daily pipeline as shown in Section 5, indicating that day-level aggregation is sufficient to capture most meaningful degradations discussed by users. While finer time bins can surface additional short-lived disruptions, many of these events span only a small fraction of the day and resolve fairly quickly using existing techniques. More broadly, these findings suggest that temporal granularity could be treated as an adaptive parameter rather than a fixed design choice. Our current implementation uses conservative thresholds based on sample counts and client diversity. Future work could dynamically adjust granularity based on statistical power—accounting not only for the number of measurements, but also their variance, consistency across users, and separation from baseline behavior. Such an approach could enable finer-grained detection when confidence is high, while defaulting to coarser aggregation when data is sparse or noisy.

B Evaluation Details

B.1 Validation Dataset Collection

We describe in more detail how we collected each of the datasets introduced in Section 5.2.

ISP Status Pages: We crawled the archive of ISP status pages by querying the Wayback Machine for snapshots of specified URLs within a given date range, filtering for valid snapshots, and using Selenium to fetch and save the rendered HTML content locally. We then manually investigated each of these files to identify whether an incident was flagged and where it was happening.

Mailing lists: The NANOG and the Outages mailing lists [67, 78] are valuable sources of operator-discussed network incidents. Posts often include detailed descriptions of affected networks, regions, and protocols. To extract this information, we used a language model to process posts since July 2024 and manually verified each extracted event to ensure that it is the type of event we intend HERMES to detect. We then manually inspected each extracted event to confirm it described a verifiable incident at a granularity that HERMES could detect (i.e., visible at an ⟨ISP, metro⟩ level, not a single customer or link).

Reddit: Since the updated API terms took effect on June 30, 2023, the process of crawling Reddit has become much harder for researchers. We rely on Arctic-Photon Reddit [77] to crawl a list of subreddits devoted to large ISPs, networking, system admins and large cities from the 1st of July 2024 onwards. By the end of this process, we get a total of 405,074 posts and 4,372,527 comments. To narrow down the dataset for further analysis, we filter for posts and comments containing specific keywords related to network issues (e.g., 'internet outage', 'slow internet', 'latency issues'). Additionally, we crawl regional subreddits for mentions of major access networks by name if they appear in a post or comment. By the end of this process, we observe a total 18,323 posts with 83,794 comments. We then process our relevant subset of Reddit posts and comments to identify Internet connectivity issues by querying OpenAI's GPT4o-mini model [71] to extracts details such as the nature of the issue, affected networks, and ISPs. We retain a Reddit thread as a candidate event only when multiple users report a similar problem affecting the same provider and location; this step filters out isolated complaints and customer-specific issues. Because Reddit posts rarely mention ASNs explicitly, we map the extracted provider to candidate ASNs using PeeringDB [76], and map the reported location to a metro area. A single thread can produce multiple candidate events if users report problems in different providers or metro areas. By the end of this process, we obtain 3,207 posts about possible events. We manually reviewed 288 of the Reddit posts and found that 213 (74.0%) described actual network events. Given the high proportion of accurate detection by the LLM, we assumed that most of the 3,207 events identified by the LLM are also legitimate network events and treat them as a reasonable approximation of ground truth.
[5]

B.2 Details on Evaluation of Recall

We use the datasets collected above to evaluate HERMES's recall.

ISP status pages: HERMES detects 85.1% of ISP-reported outages.

Mailing lists: From 42 manually verified incidents, HERMES successfully detected 31 (73.8%). HERMES missed 9 events because it did not identify the affected user group as experiencing an issue and 2 events because no anomalous paths were observed crossing the provider in question.

Reddit: Within the 213 manually verified events reported on Reddit, 138 (64.7%) were observed by HERMES. Across the full dataset of 3,207 possible events reported on Reddit, HERMES detected 57.2%, with the rest either missed by HERMES or misidentified by the LLM parsing of Reddit. Extracting meaningful signals from Reddit remains challenging due to variations in user expertise and the possibility that some disruptions occur at a finer granularity than our metro-level analysis. Despite these limitations, HERMES's detection rate significantly surpasses those of Cloudflare and IODA, which capture fewer than 5.2% of the events—11× less than what we had.

[5]A Reddit post maps to a HERMES event if several users report degradation in the same AS/metro in the same day. A single Reddit thread may correspond to multiple HERMES events if users in different regions and ASes are affected.

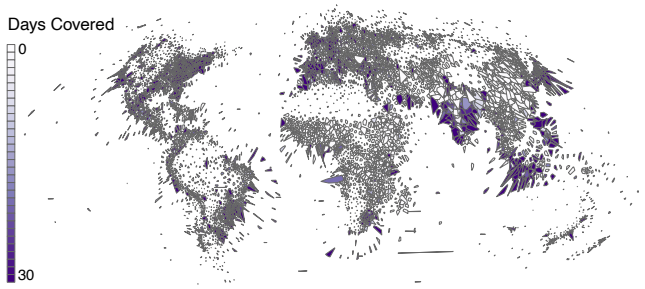


Figure 10: Anamorphic map showing the number of days with sufficient network coverage across metro areas. Each cell corresponds to metro and is scaled by population. Color indicates, for each metro, the number of days in the month for which the best-covered ISP in that metro has enough measurements to run HERMES.

B.3 Further Evaluation of Coverage

In Section 5.3, we centered our discussion on HERMES's coverage of user-hosting networks. This section examines (i) metro and (ii) infrastructure-level visibility.

B.3.1 Metro Coverage of HERMES. We compare HERMES's measurement coverage over a 30-day period with the population size of metro areas. We use satellite-derived population estimates [99] together with iGDB Voronoi cells [2] to delineate metro boundaries and aggregate population totals. For each metro, we identify the ISP with the largest number of days satisfying HERMES's minimum measurement requirements (i.e., 25 measurements from at least 5 different source IPs) and use that number as the metro's coverage score. A score of 0 means that no ISP in the metro had sufficient measurements on any day in the 30-day period and a score of 31 that at least one ISP had sufficient measurements for all days.

We visualize this relationship using an anamorphic map (Figure 10). Unlike traditional geographic maps that preserve the physical size of regions, an anamorphic map adjusts the size of each region proportionally to a chosen metric—in this case, population. This allows us to emphasize regions where Internet performance issues could have impact on the most users. This visualization is particularly useful for identifying regional biases or gaps in HERMES's deployment. For example, regions with large cells but lighter colors suggest that HERMES's measurement density does not align with the population distribution, potentially leaving key metro areas underrepresented. Conversely, darkly colored large cells signify strong and consistent coverage in populous regions, underscoring areas where HERMES effectively monitors network performance. Our analysis shows that coverage is strong in the most populous regions: 71.5% of the top 10% of metros by population (and 100% of the top 1%) have continuous coverage over the full month, and 87.5% have coverage for at least 15 days. Coverage remains sparse in several high-population regions where the number of speed tests is low relative to the underlying population (e.g., Russia, China, parts of India, and parts of Africa).

B.3.2 Infrastructure Coverage of HERMES. Our coverage goal is to ensure that HERMES can detect events that significantly impact end-users. Even if HERMES lacks measurements from a user group, it

can potentially detect events impacting the user group if it monitors parts of paths to/from the user group via measurements on overlapping paths from other user groups.

To quantify this *path-level* coverage, we evaluate the fraction of infrastructure elements traversed by HERMES measurements, including metros, PoPs, ASes, and IXPs. We use iGDB [2] to identify PoPs defined as $\langle AS, metro \rangle$ pairs and compare our observations against CAIDA's Internet Topology Data Kit (ITDK) [10]. ITDK aggregates large-scale traceroute measurements collected from dedicated probing infrastructures to all /24 prefixes across the Internet, applies router alias resolution, and maps observed interfaces to ASes, IXPs, and geographic locations to infer a macroscopic view of Internet topology. While ITDK is expected to observe more entities by design, this comparison provides useful context for the scope of HERMES's visibility. Table 5 shows that HERMES observes traffic across 12,048 ASes, 4,492 metros, and 323 IXPs, spanning 239 countries. Although this represents a smaller fraction of total infrastructure than ITDK, the geographic reach of HERMES is comparable: user-driven measurements intersect nearly every region of the Internet. Moreover, HERMES covers 55.6% of all observed $\langle$AS, metro$\rangle$ pairs, indicating substantial overlap with the access and transit networks that carry end-user traffic. Despite covering only 17.7% of ASes, HERMES observes paths through infrastructure that is likely to carry traffic for a large fraction of Internet users, because user demand is concentrated in a relatively small set of access/transit networks and metro interconnections (§5.3). These results highlight that HERMES provides a unique lens on how Internet infrastructure behaves under stress, making it especially suited for detecting, diagnosing, and contextualizing user-impacting events.

	Viewed by HERMES	ITDK	Whole Internet
Count of IXP	323 (26.8%)	614 (50.9%)	1205
Count of $\langle ASN, IXP \rangle$	4,681 (6.9%)	18,044 (26.6%)	67,797
Count of ASN	12,048 (17.7%)	67,620 (87.1%)	77,642
Count of Metro	4,492 (61.2%)	5,312 (72.3%)	7,342
Count of Country	239 (96.0%)	248 (99.6%)	249
Count of $\langle ASN, Metro \rangle$	56,899 (55.6%)	76,756 (75.1%)	102,205

Table 5: Comparison of Internet entities viewed by HERMES, ITDK, and the whole Internet. Percentages are relative to the whole Internet.

B.3.3 Representativeness of Paths Toward M-Lab. Because HERMES relies on speed tests to M-Lab servers, a natural question is whether paths toward M-Lab are representative of the paths that carry a large fraction of Internet traffic. To examine this, we use RIPE Atlas [83] traceroutes toward popular destinations as a point of comparison. Specifically, we consider traceroutes to IP addresses hosted by large CDNs according to prior literature [34] and to the top 10,000 websites in the Chrome UX Report (CruX) [35], which together account for a substantial share of global Internet demand. We analyze five days of all RIPE Atlas traceroute measurements to these destinations, leveraging the fact that RIPE Atlas already probes popular services extensively as part of its regular operation. This step allows us to observe a broad, diverse set of commonly used paths without launching additional measurements ourselves; in contrast, a bespoke RIPE campaign would have yielded substantially lower coverage due to credit restrictions. In total, we recovered 12,274,011 traceroutes originating from 13,284 different probes in 2656 ASes to 171 ASes. From these traceroutes, we extract the set of ASes, metros, IXPs, $\langle AS, metro \rangle$ pairs, $\langle AS, IXP \rangle$ pairs, and directed AS-level edges, and compare them to the corresponding sets observed by HERMES along paths to M-Lab. We find substantial overlap across all granularities: HERMES covers 80% of the ASes observed in RIPE Atlas (4,321/5,383), 67% of metros (2,683/4,032), and 71% of $\langle AS, metro \rangle$ pairs (12,181/17,070). At interconnection points, HERMES observes 46% of RIPE-seen IXPs (202/436) and 48% of $\langle AS, IXP \rangle$ pairs (1,212/2,501). At the AS-level edge granularity, the two datasets overlap on 7,258 directed inter-AS links, corresponding to 28% of the RIPE observed edges.[6] While paths from RIPE Atlas probes toward popular destinations naturally expose a broader tail of unique links (as expected from Internet flattening), the consistent overlap across the other segments indicates that paths to M-Lab traverse much of the same core, transit, and interconnection infrastructure used by high-volume Internet paths.

B.4 Sensitivity Analysis

We evaluate the robustness of HERMES by systematically sweeping conservative ranges for its main hyperparameters: majority thresholds, latency and throughput sensitivity thresholds, baseline window lengths, and minimum coverage requirements. The analysis is performed over a week of HERMES's run in December 2024, chosen to capture realistic day-to-day variability while keeping the sweep computationally tractable. For each sweep, we measure how the total number of detected events varies relative to the paper's default configuration. Across all dimensions, event counts change smoothly and monotonically: relaxing thresholds increases detections, while tightening them decreases detections, without sharp discontinuities, phase transitions, or unstable regimes. The qualitative conclusions of the paper (e.g., relative prevalence of latency vs. throughput events and coverage trends) remain unchanged across a broad region of the parameter space. The default configuration used throughout the evaluation lies in a flat region of this space, where moderate parameter changes lead to proportionally small changes in outcomes, indicating that results are not driven by finely tuned choices. Furthermore, the structure of our pipeline naturally lends itself to automated hyperparameter tuning or operator-driven calibration in future deployments, allowing thresholds to be adapted to different measurement densities, operational goals, or risk tolerances without changing the underlying methodology.

B.5 Impact of Omitting Forward or Reverse Paths

To quantify how source attribution depends on path visibility, we run HERMES's localization pipeline on the same set of detected events under three configurations: using only forward paths (server-to-client traceroutes), using only reverse paths (client-to-server traceroutes), and using both directions together. All other components of the system—including event detection, topology construction, and temporal and correlation tomography—are held constant. Any

[6] For this analysis, we exclude the final hop into hypergiant networks, since this link is inherently unobservable from our measurements and including it would artificially understate our coverage.

differences in attribution therefore arise solely from the availability of forward and/or reverse path information.

We consider the set of events for which HERMES (running with all data) identified a source. Using only forward paths, HERMES can only attribute 24.6% of the events; using only reverse paths yields a comparable 24.9%. In contrast, 50.5% of the events can be attributed only when both forward and reverse paths are available. These cases typically correspond to asymmetric routing changes or congestion that manifests predominantly in one direction, where single-direction visibility provides an incomplete view of the path. As a result, systems that observe only one direction systematically miss or misattribute a substantial fraction of events.

These trends are consistent across latency- and throughput-driven events and across regions, indicating that the need for bidirectional visibility is not driven by a small number of outliers but reflects a general property of Internet-scale routing. Together, these results show that while forward and reverse paths each provide partial attribution power, accurate localization at Internet scale requires visibility into both directions.

B.6 Reverse Traceroute Evaluation

To reappraise the performance of the reverse traceroute system, we analyze two metrics over a six-month period: (i) the percentage of reverse traceroutes successfully measured and (ii) the percentage of reverse traceroutes that are trustworthy. When reverse traceroute is unable to measure a hop along the path, it assumes the next hop is traversed symmetrically and then continues measurement from the next hop. Previous analysis found that this assumption is more likely to be wrong when it crosses an interdomain boundary, and so any reverse traceroute including such an assumption is flagged as untrustworthy [97].

For (i), the success rate of reverse traceroutes varies between 52% and 66% across different days, with most days averaging around 58%. For (ii), 58% to 67% of *successful* measurements (approximately 35% of all *attempted* reverse traceroutes) do not rely on assuming interdomain symmetry, meaning they can be fully trusted. The most recent published evaluation of reverse traceroute reports quantities that can be mapped onto the same two metrics [97]. The study ran 101M paired forward and reverse measurements and obtained complete forward traceroutes for 57M paths, or roughly 56% of attempted measurements. Under our current response definition, this corresponds to metric (i), since the current system returns a reverse-traceroute response at least whenever the corresponding forward traceroute succeeds. Among these forward-complete paths, the study obtained complete reverse traceroutes for 31M paths. Since RevTr 2.0 avoids returning reverse paths that require interdomain symmetry assumptions, these 31M paths correspond to trustworthy responses under metric (ii), yielding a trusted-response rate of roughly $\approx 54\%$. Overall, compared to the IMC deployment, the current system has a similar response rate for metric (i), approximately 58% versus 56%, and a higher trusted-response rate for metric (ii), 58-67% versus roughly 54% under the same no-interdomain-symmetry trust criterion.

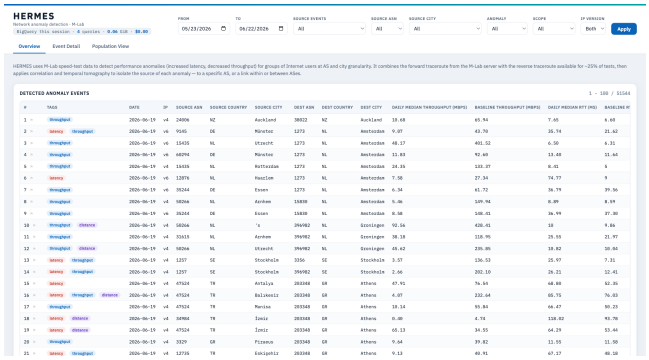


Figure 11: HERMES dashboard aggregate *Overview*. A filterable, sortable table lists the daily anomalies with their source/destination AS, city, and country, the observed throughput/RTT deviation, and—when localized—the network segment attributed as the source. A world map and an entity bar chart (below the table) rank the networks and metros that explain the most events over the selected window.

B.7 Number of Speed Tests during Events

To test whether users are more likely to run speed tests when experiencing degraded performance, we examine post-hoc whether user groups show increased measurement activity on days flagged as anomalous by HERMES. For each user group over a three-month period, we compare the number of tests on anomaly days to non-anomaly days using a one-sided Welch's t-test. Among user groups that experienced at least one anomaly, 60.2% show a statistically significant increase in test volume on anomaly days (one-sided t-test, $p < 0.05$). An additional 35.4% exhibit a positive but non-significant increase, with $t-$values below the significance threshold of 1.96. Only 4.4% of user groups show fewer tests during anomaly days than on average days. This result addresses a potential concern with user-driven measurements: HERMES could miss degradations if users generated fewer measurements precisely when performance was poor. We find the opposite pattern for monitorable user groups. Conditional on a user group having enough measurements to be evaluated, anomalous days usually have higher test volume than non-anomalous days. Thus, while sparse user groups and regions can still limit coverage, the days on which HERMES observes degradations are not typically days with reduced measurement activity. This suggests that same-day measurement availability is unlikely to be the main reason HERMES misses events among covered user groups. At the same time, increased test volume is not itself sufficient evidence of a degradation. We also observe spikes in test volume that do not coincide with performance degradations, suggesting that volume alone would produce false positives and must be combined with performance-based detection.

C Complementary Measurement Studies with HERMES

C.1 Dashboards

We acknowledge that some of the conclusions drawn by HERMES may remain unverifiable; HERMES should therefore provide transparency into why a particular anomaly is labeled and how it was attributed to a specific network component. To support this, we built a public web dashboard, accessible at `https://hermes-dashboard.`

org/, that exposes every flagged event together with the measurements and reasoning behind it. The dashboard is organized into coordinated views that share a common set of filters—date range, source ASN/city/country, anomaly type and intra-/inter-domain scope, an aggregate *Overview*, a *Population View* summarizing the selected window, and a per-event *Event Detail* drill-down.

The Overview (Figure 11) lists the detected events in a filterable, sortable table, including the affected source/destination ASNs, cities, and countries; the performance deviation observed (e.g., increased latency or reduced throughput); and, when available, the isolated source of the issue along the path. Each event is tagged with whether it is intra- or inter-domain and whether the responsible segment lies on the forward or reverse path. A world map and an accompanying bar chart (below the table) rank the ASes and metro regions that explain the largest number of events over the selected window. The Population View complements this with statistics—event counts, severity (throughput-drop and RTT-increase) percentiles, the fraction of inter-domain and root-caused events, a per-day severity trend, and distributions of throughput drop, RTT delta, and event class.

The Event Detail view lets operators and researchers drill into a single anomaly affecting a user group on a given day and inspect the traceroutes and speed tests behind the inference. It provides several coordinated visualizations: (i) RTT and throughput time-series plots showing each measurement relative to its baseline, with the event day highlighted; (ii) detection statistics that compare the baseline and day-of distributions and report the corresponding statistical test (e.g., Mann–Whitney and Wasserstein); (iii) a *temporal tomography* view (Figure 12a) that contrasts the user group's usual, healthy-baseline AS-metro path with the path observed on the day of the event, accompanied by a per-segment table that quantifies how traffic shifted and labels each link as *abandoned* or *diverted onto*; (iv) a *correlation tomography* AS-metro topology graph (Figure 12b), laid out either hierarchically or force-directed, in which edges crossed by anomalous paths are highlighted and edge width encodes the number of traversals; (v) logical path tables listing the IP, AS, organization, city, and per-hop RTT of each hop on the forward and reverse paths; and (vi) forward and reverse geographic maps of the physical route. Users can click on individual nodes, links, or measurements to filter the view and inspect the associated metadata and traceroutes. Finally, the interface includes a lightweight feedback mechanism—for instance, flagging an incorrect hop geolocation or a misattribution—so that operators can submit corrections that feed back into HERMES's event attributions.

C.2 HERMES in Numbers

To contextualize the scale and operating regime of HERMES, we report monthly aggregate statistics computed from daily runs over December 2025. (Table 6). On average, HERMES processes 30.5M NDT speed tests per day and monitors roughly 28K user groups with sufficient measurements. Of these monitorable user groups, about 980 experience a detectable performance degradation on a typical day, corresponding to on average 3.4% of all covered user groups. Detected events are predominantly short-lived, with the majority lasting a single day, while a small tail persists for multiple days. HERMES attributes these events to several hundred distinct network segments per day, split roughly evenly between interdomain and intradomain infrastructure.

Metric (averaged across a month)	Value
NDT speed tests per day	30.5M
User groups with sufficient coverage per day	27,789
Impacted user groups per day	1,040
Total events per day	1,004
Distinct responsible segments per day	442
Interdomain segments	235
Intradomain segments	207
Events lasting 1 day	759
Events lasting 2 days	138
Events lasting 3 days	54
Events lasting 4 days	33
Events lasting ≥5 days	19
Throughput-based event rate (daily avg.)	2.96% (of all user groups)
RTT-based event rate (daily avg.)	0.79% (of all user groups)

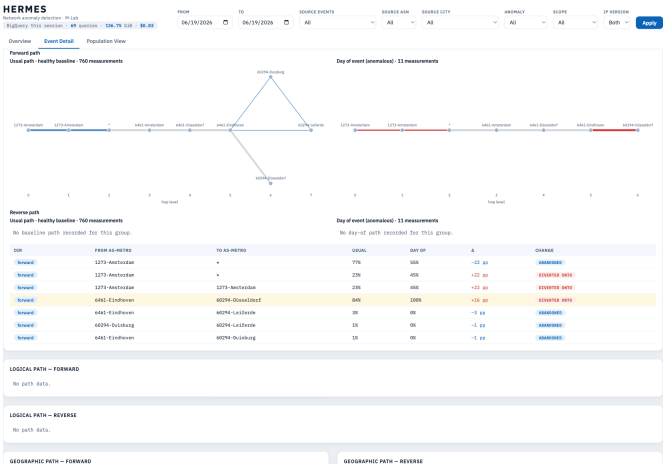
Table 6: Monthly summary statistics for HERMES (December 2025). Values are averages per day unless otherwise noted.

C.3 Quantifying Routing Asymmetry

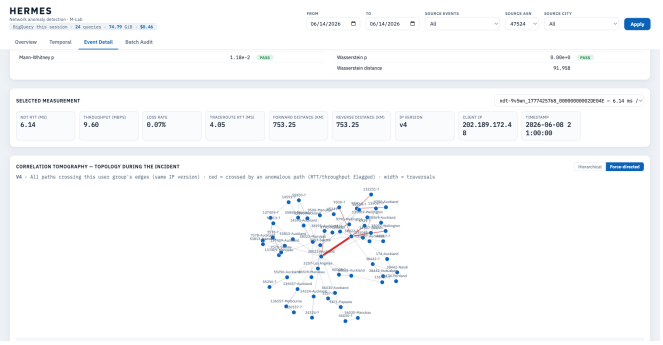
In Figure 13, for each measurement from a user group experiencing an anomaly, where the end-to-end metric exceeds the baseline, we compare the relative lengths of the forward (server-to-user) and reverse (user-to-server) paths. Specifically, we compute the asymmetry ratio as the ratio of the forward path length to the reverse path length. For visualization, if the forward path is more circuitous (i.e., the ratio is less than 1), we invert the ratio and assign it a negative value. This metric allows us to determine whether routing inefficiencies are primarily occurring in the forward path (red) or the reverse path (blue). Our analysis shows that, for approximately 72% of paths, the reverse path is longer than the forward path. Additionally, in about 10% of cases, the reverse path is at least twice as long as the forward path. This finding aligns with recent research indicating that optimizing reverse paths is generally more challenging [51, 102].

C.4 Persistent Congestion is Persisting

To identify congestion events, we examine paths that remain unchanged before and during performance degradation and use our correlation tomography algorithm to determine which network component is the likely source of the issue. Table 7 highlights the top interconnections most frequently implicated in congestion events detected by HERMES. These interconnections are ranked by the frequency with which they are identified during anomaly detection, providing insight into which links are commonly associated with degraded performance. These interconnections often involve large transit providers. Prior work on persistent congestion relied on running extensive active measurements that required to pre-select links or locations to probe [25], whereas HERMES leverages existing data to detect potential congestion points and supplements them with a few targeted traceroutes only when needed. This approach eliminates the need for specialized probe deployments and ensures



(a) Temporal tomography: the user group's usual path (left) vs. the path observed on the day of the event (right), with a per-segment table labeling each link as *abandoned* or *diverted onto*.



(b) Correlation tomography: the AS-metro topology of all paths crossing the user group, with edges traversed by anomalous (RTT/throughput-flagged) paths highlighted in red and edge width encoding the traversal count.

Figure 12: HERMES's per-event *Event Detail* drill-down. 1. Temporal tomography contrasts the healthy-baseline path with the day-of-event path to localize the responsible segment; (b) the correlation-tomography topology graph shows which AS-metro links the anomalous paths share. The same view also provides RTT/throughput time series, statistical-test verdicts, logical-path tables, and forward/reverse geographic maps (not shown).

that the focus remains on user-impacting congestion observed in the wild.

C.5 Weather and Cable Cuts

This section demonstrates HERMES's ability to analyze and visualize metro-level network anomalies associated with weather phenomena and infrastructure disruptions. We focus on a small set of high-impact events—including floods, hurricanes, typhoons, and a subsea cable cut—and analyze network behavior before and after each event. For each case, we compare the fraction of ASes experiencing anomalies in impacted metros against a pre-event baseline, isolating event-driven effects from normal variability (§6).

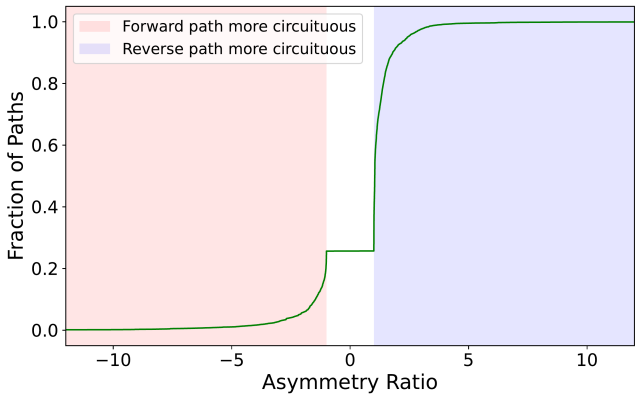


Figure 13: Distribution of asymmetry ratio for anomalous measurements. Negative values indicate the forward path is more circuitous, while positive values indicate the reverse path is more circuitous. For 72% of paths, the reverse path is more circuitous. For about 10% of the paths, the reverse path is at least twice as circuitous as the forward path.

Table 7: Persistently congested interconnections identified by HERMES. Each row lists an AS–metro link that was repeatedly selected by correlation tomography as the likely source. Links are ranked by the number of anomaly days in which they were implicated.

Rank	Interconnection	Organizations
1	4755-Mumbai-IN - 9498-Mumbai-IN	Tata - Bharti Airtel
2	1273-Milan-IT - 3356-Milan-IT	Vodafone Group - Level 3
3	3320-Frankfurt-DE - 3356-Frankfurt-DE	Deutsche Telekom - Level 3
4	3356-Rome-IT - 6453-Milan-IT	Level 3 - Tata
6	1221-Brisbane-AU - 7575-Sydney-AU	Telstra - AAPT
7	3320-Frankfurt-DE - 3356-Frankfurt-DE	Deutsche Telekom - Level 3 (Lumen)
8	17676-Tokyo-JP - 2518-Tokyo-JP	Softbank Corp. - KDDI
9	3356-Rome-IT - 6453-Milan-IT	Level 3 - Tata
10	1267-Rho-IT - 6453-Milan-IT	Wind Tre S.p.A - Tata

Valencia flooding (October 2024): Severe flooding in eastern Spain produced the strongest and most spatially extensive signal we observed. Many metros in and around Valencia exhibited large increases in the fraction of anomalous ASes (often exceeding 0.4), with effects extending beyond the immediate flood zone. This pattern is consistent with shared infrastructure dependencies and regional rerouting during the disruption.

Typhoon Shanshan, Japan (August 2024): In contrast, the impact of Typhoon Shanshan was limited and highly localized. Only small increases in anomaly fractions appeared in a few metros along the storm's path, while most of Japan remained near baseline. This suggests that network hardening and routing diversity mitigated widespread performance degradation.

Hurricane Milton, United States (October 2024): Milton's passage through Florida and the southeastern U.S. coincided with elevated anomaly fractions concentrated along its track. We also observe scattered increases in more distant metros, suggesting secondary effects from upstream dependencies or traffic shifts beyond the directly affected region.

Baltic Sea subsea cable cut (October 2024): The cable disruption produced a sharply localized signal: anomaly increases were concentrated near the Finnish landing point, with little change elsewhere. The limited spatial footprint suggests that the affected cable carried only a small fraction of the traffic visible to HERMES.

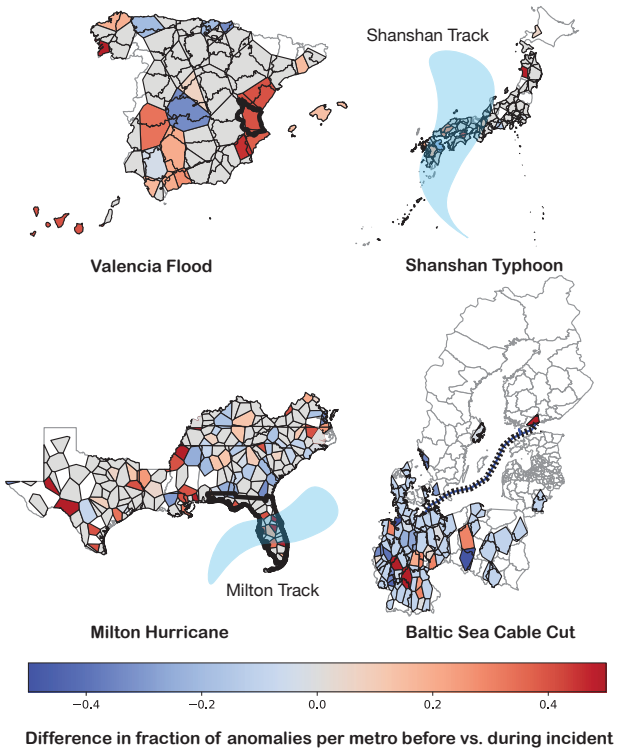


Figure 14: All events studied in Appendix C.5.