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

Loqman Salamatian[⋆,§] *Kevin Vermeulen*[†] *Dave Choffnes*[‡] *Ethan Katz-Bassett*[⋆] *Phillipa Gill*[§]

[⋆] *Columbia University* [†] *LIX-CNRS* [‡] *Northeastern University* [§] *Google*

Abstract

Internet observatories are essential for monitoring network health and performance. Diagnosing performance degradation and pinpointing its 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 system currently provides this capability. Existing solutions rely on coarse-grained signals that fail to capture end-user performance, while proprietary solutions, in addition to being inaccessible, 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 never been used to automatically detect and explain end-user performance degradations at scale. To achieve these goals, HERMES combines robust 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 on sparse public data, HERMES attains precision comparable to a proprietary CDN system and surfaces 12× more publicly discussed events than existing public observatories. We demonstrate HERMES's ability to (i) track the impact of weather and cable cuts, (ii) diagnose routing inefficiencies, and (iii) identify persistently congested links.

1 Introduction

The Internet is a critical infrastructure, with applications from online gaming to remote health care relying not only on availability, but also good performance. However, existing public Internet observatories mainly focus on outages, shutdowns and censorship [24, 42, 69, 85]—whereas harmful performance degradations (*e.g.,* increased latency) remain largely invisible to stakeholders. Instead, today, that visibility is confined to private monitoring services run by cloud providers or 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 can support faster diagnosis, more credible postmortems, and coordination across organizational boundaries—without relying on access to proprietary telemetry. From our experience working with CDNs and ISPs, resolving these issues often requires collaboration across networks, but in the absence of common measurement signals, such collaboration can be hard. Improved visibility can also inform broadband funding decisions (*e.g.,* via the BEAD program [64]) and guide resilience and performance improvements.

To close this gap, we introduce HERMES, a system that continuously detects, contextualizes, and localizes user-facing performance issues using open data from end-user measurements. Achieving this at Internet scale appears to require two infeasible capabilities: (i) continuous performance monitoring at Internet scale and (ii) complete bidirectional maps of Internet paths. HERMES sidesteps these requirements by leveraging publicly available speed test results, paired forward and reverse traceroutes, statistical anomaly detection, and tomography-based localization, showing that comprehensive visibility is attainable with existing data and targeted measurement.

Repurposing speed test data to detect performance degradation: Our starting insight is that NDT speed test data [61] can be repurposed to systematically identify performance degradations and, when combined with topology measurements, localize where they occur. Google search results for terms like "test my Internet' ' include a widget to perform an NDT speed test directly from the search result page (a screenshot is shared in Figure 8), making it one of the most widely used performance measurements, with approximately 4 million tests performed daily across diverse locations and networks. The test data is published publicly by M-Lab (§5.3). Unlike other speed test platforms that often measure performance within the user's ISP's network [21, 57], NDT speed tests are specifically designed to assess performance across peering interconnections, frequent points of congestion [28, 32, 53]. This design ensures that mea-

surements traverse more networks, providing greater insight into Internet-wide performance. Importantly, the vast majority of these tests are initiated by users themselves, often in response to perceived connectivity issues, making the dataset highly reflective of moments where users are most interested in network performance. NDT pairs tests with traceroutes and reverse traceroutes, providing bidirectional visibility into network paths [93]. 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 the automated identification of statistically significant degradations in network performance (e.g., increased latency or decreased throughput) that persist for an extended period 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 [53], users' Internet plans [75] and accessibility [74], Starlink's network performance evolution [63], or Venezuela's political crisis [15]. 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 have prevented automated, large-scale event detection from M-Lab data: (i) speed tests capture localized performance and can be distorted by customer-side issues, making it difficult to distinguish between last-mile problems and broader network disruptions; (ii) tests are user-triggered, creating irregular sampling; (iii) the users initiating these tests 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 then localizes their likely sources through a novel tomography-inspired method that leverages bidirectional path measurements and distinguishes between performance degradation induced by routing changes that shift traffic onto worse-performing paths and congestion occurring along stable routes (§§ 4.2 and 4.3). HERMES operates at a daily granularity (matching operational practice, reducing the effect of ephemeral issues and following M-Lab's own data publication cadence) and has processed 4 billion speed-tests over a 5-month period, detected 65K network events affecting 37K $\langle \text{AS}, \text{metro} \rangle$[1] pairs across 9,710 ASes in 166 countries. Compared to BlameIT [47], a proprietary tool used by a major cloud provider, 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 it achieves sufficient coverage to detect degradations affecting between 60-80% of the Internet user population in Europe and North America, and at worst 40-70% in Africa (§5.3). Its use of bidirectional path measurements is key to accurately diagnosing network issues (§5.4). This work presents minimal ethical concerns, though we address potential implications in Appendix A.

[1] Here metro means metropolitan area.

2 Related Work

Our goal is to develop an observatory that detects performance degradations impacting end users. Existing approaches fall into two categories, each with limitations:

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 [9, 29, 33, 42, 49, 52, 58, 77, 97]. BGP provides coarse visibility, but most announcements reflect routine churn [41], and transient congestion often leaves no BGP trace (§3). Active probing (e.g., ping, traceroute) is widely used to map paths and confirm reachability, adding context beyond passive or user-driven measurements. However, systems built on this approach, such as IODA [42], focus on detecting outages and responsiveness rather than end-user performance degradation. 007 [5] localizes failures in highly structured datacenter networks, a setting that differs fundamentally from our goal of identifying user-visible performance issues at Internet scale. In contrast, HERMES uses measurements from M-Lab servers of application-level performance paired with forward and reverse traceroutes, enabling visibility into both user performance and bidirectional paths.

Prior works also highlight fundamental challenges in inferring congestion from end-to-end measurements, especially speed tests. Luckie et al. [54] detail the ambiguity of attributing interdomain congestion from latency measurements due to alias resolution, router ownership, and path asymmetry. Sundaresan et al. [91] extend these concerns to throughput inference on M-Lab data, showing that simplifying assumptions—such as direct AS-level connections between servers and clients—often fail. They also note biases from crowdsourced sampling, home-network variability, and unclear congestion thresholds. HERMES addresses these challenges by pairing each test with forward and reverse traceroutes to capture paths, grouping measurements by network and location to reduce noise, and applying correlation-based tomography to pinpoint likely degradation points. When uncertainty remains, targeted probing expands path coverage and improves attribution precision.

Other systems illustrate these difficulties even in more controlled settings: CableMon [37] combines RF modem data with operator tickets yet struggles to isolate last-mile noise, while Jitterbug [16] derives RTT metrics from jitter dispersion but depends on dense, continuous probing. Although the accuracy of NDT to estimate bandwidth has been questioned

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[97]	007[5]	BlameIT[47]	Cloudflare Radar[24]	CEM[19]
End-User Perspective	✓	**x**	✓	**x**	✓	✓	✓
Performance Metrics	✓	**x**	**x**	**x**	✓	✓	✓
Pinpoint Causes	✓	**x**	✓	✓	✓	**x**	✓
Bidirectional Paths	✓	**x**	✓	**x**	**x**	**x**	✓
Open Data	✓	✓	**x**	✓	**x**	**x**	**x**
Maintainability (discussed in Appendix E)	✓	✓	**x**	–	–	–	**x**

[57] and its methodology has evolved (e.g., MSAK [60]), these concerns are tangential to our approach: HERMES uses throughput only as a *relative performance signal*. By comparing temporal shifts within stable user groups (defined more formally in Section 4.1.1), it robustly detects degradation even if absolute test values fluctuate.

Privileged vantage points focusing on performance signals: The second category of related work comprises systems built on privileged vantage points, typically operated by cloud providers, CDNs, or commercial monitoring platforms. These systems have access to dense, centrally orchestrated measurements and global traffic visibility, which enables them to detect and localize problems with relatively simple statistical methods. However, their localization typically stops at the network (AS) level, and their datasets are proprietary and closed to the research community. By contrast, HERMES advances the state of the art by enabling Internet tomography at finer levels of granularity. Operating exclusively on open, user-driven data, which is sparse, noisy, and unevenly distributed, HERMES can attribute degradations not only to entire networks but also to specific metropolitan areas or even to particular peering links. This flexibility allows us to detect and localize a broader class of events: from large-scale outages that span entire networks, to metro-level disruptions, to congestion isolated to a single interconnection. Achieving this precision under open-data constraints requires new statistical inference techniques that go beyond what privileged-vantage-point systems had employed.

For example, Cloudflare Radar [24] identifies outages by analyzing aggregate traffic volumes but does not capture how these events affect end-user performance. BlameIT [47], deployed at Microsoft, localizes issues at the AS level using a large amount of regularly scheduled traceroutes and measurements of end-to-end user connections to Microsoft services. Its approach relies on the assumption that latency noise (e.g., queueing delays) can be smoothed out through frequent, uniformly distributed probes—a condition that holds in cloud-scale deployments but not in our setting, where measurements are user-driven, unevenly distributed, and often too sparse for such averaging.

Older systems such as PlanetSeer [97] and CEM [19] are no longer available and targeted narrower settings: PlanetSeer, built on PlanetLab [20], detected path anomalies between research nodes. Unlike HERMES, it did not target crowdsourced clients, incorporate user-group baselines, or attempt metro- or interconnection-level attribution. CEM integrated application metrics to detect BitTorrent-specific network events, but its focus has become less relevant as BitTorrent usage has declined. Ookla's Open Data [68] and Cloudflare Speed Test [23] provide data only at coarse-grained geolocations, without path information. Crowdsourced incident platforms (e.g., Downdetector [30] 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 provide visibility into underlying network paths or causes. Moreover, their detection methodologies are neither transparent nor peer-reviewed, which limits their utility for operational diagnosis.

Table 1 synthesizes the systems discussed in this section, spanning academic efforts, commercial platforms, and congestion-inference studies. We compare them on six dimensions—end-user perspective, performance metrics, attribution capability, bidirectional path visibility, openness, and long-term maintainability. This summary highlights that no prior system provides open, path-aware, performance-focused observability at Internet scale, a gap HERMES fills.

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 (Cloudflare Radar, IODA [22, 43]) but drew over 100 Reddit comments [3]. This example demonstrates the types of challenges our system addresses and previews how our methodology overcomes them.

Figure 1 shows round-trip latency and throughput measurements from NDT speed tests between Cogent (AS174) users in Chicago and an M-Lab server hosted in Chicago by TATA (AS6453). On July 4th, at around noon, the latency increases and the throughput sharply decreases, signaling a performance degradation.

A single speed test initiated by a user at that moment would reveal only that this particular user seems to be experiencing degraded connectivity. Without additional context, this observation offers no evidence that the problem generalizes to other

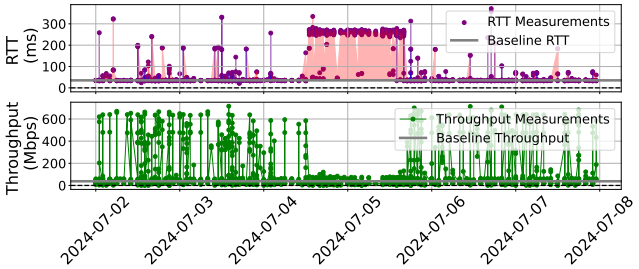


Figure 1: Round-trip latency (top) and throughput (bottom) between Cogent customers (AS174) in Chicago and a server hosted by TATA (AS6453) in Chicago for 5 days between 2024-07-02 to 2024-07-07. The baseline is the median RTT (top) and throughput (bottom) observed across a period of 7 days before the window of interest.

users. In fact, even under normal conditions, throughput measurements often include a mix of high and low values—some tests show poor performance for reasons unrelated to network issues (e.g., faulty Wi-Fi router or background applications). As a result, a few bad results alone are not enough to indicate a broader event. However, by examining multiple speed tests from many users served by the same network in the same location to the same destination, we observe a simultaneous increase in latency and decrease in throughput—suggesting that the problem is deeper in the network path.

While these aggregated performance metrics reveal a widespread event, they do not explain its source. 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 some irregularities: although the AS-level path between Cogent and TATA remained stable (*i.e.,* direct interconnection), traceroute probes show that the geographic path (Figure 2) was rerouted through distant interconnection points such as one in Texas, suggesting that a disruption in the existing interconnection point in New York has caused a shift onto more circuitous paths. Still, this explanation alone falls short of accounting for the magnitude of the observed latency increase (more than 210 ms observed versus $\approx$ 30 ms incurred by detouring through Texas). Adding reverse-path measurements, from the users to the M-Lab server, we observe traffic detours on the reverse paths stretching as far as Singapore, before reaching the server in Chicago. Combined, the detours in both directions explain approximately 190 ms of the observed latency. No traceroutes traverse the usual Chicago/NYC peering points that day, indicating that these interconnections might have gone down and traffic now relies 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.

This example points to key goals of HERMES: to both detect that performance has degraded and also to identify the network entities responsible—whether by incident or by effect. In this case, the primary trigger appears to be the disappearance of peering links in Chicago and NYC between Cogent

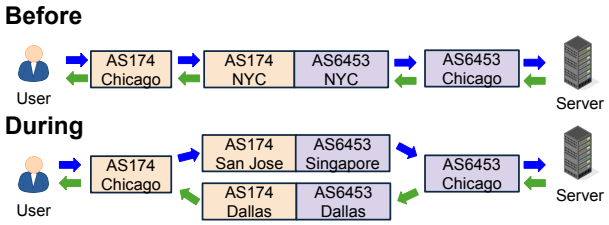


Figure 2: Illustration of bidirectional path changes during a network incident. The top diagram depicts the regular, pre-incident paths between a user and the server, passing through ASes 174 and 6453 in Chicago and NYC. The bottom diagram shows an example of paths observed during the event, where traffic is rerouted through AS6453 in Dallas and San Jose and via Singapore.

and TATA. 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., loss of 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 [3].

4 Methodology

The example above highlights key challenges we need to address, 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 C.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) routing changes that alter the path, 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 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 intrinsically 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). Each box in the figure corresponds to a stage in this process.

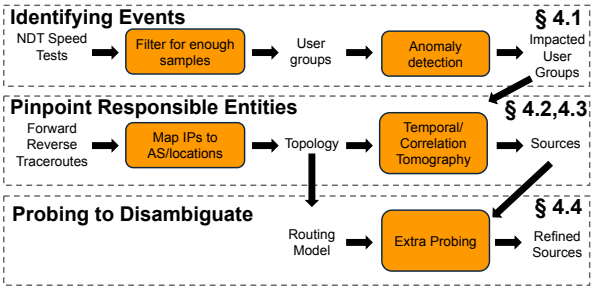


Figure 3: Schematic representation of HERMES.

4.1 Identifying Events

HERMES first identifies when a user group experiences a statistically significant degradation—an *event*—defined as a full-day deviation in throughput or latency from its historical baseline. In principle, our methodology could operate at finer time scales, but we adopt daily granularity to align with M-Lab's operational model, where test data is published in daily batches. This choice also provides statistically robust detection given the large number of speed tests collected each day. The trade-off is that short-lived or self-resolving disruptions (e.g., diurnal congestion or bursty one-off events) may be missed, though the design matches the timescales at which operators typically diagnose systemic issues [52]. Moving to sub-daily resolution would require non-trivial engineering changes to M-Lab's data pipelines, which we are actively discussing with the platform.

We detect events by identifying significant deviations in performance metrics. Speed test data is ideal because (i) it directly measures throughput and latency, and (ii) users often run tests *because* they are experiencing poor performance, introducing a *positive selection bias* [10] that concentrates measurements around real degradations, making them easier to detect.

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 exclude packet loss as a detection signal because it is is noisier than other signals and largely redundant with throughput degradation (more in discussion in Appendix B.6).

4.1.1 User groups

We define user groups (**UGs**) as pairs of $\langle \text{AS}, \text{metro} \rangle$, following common practices in network operations [13, 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 C.5 and show in Section 5.3 that HERMES encompasses a large number of networks and metros worldwide. To ensure accurate grouping, we validate client locations using multiple geolocation sources and exclude tests with conflicting or implausible location data (details in Appendix B.1).

4.1.2 Latency

We first detect performance deviations using latency, as computed by the server's transport protocol (i.e., BBRv1 [14]) over the entire test. 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 prevents any single window from disproportionately affecting the baseline latency. Because speed tests are noisy by nature, we remove clear outliers where the latency is above $5,000$ ms (less than 0.001% of our measurements).

To detect latency anomalies on a given day, we apply Welch's t-test and the Mann-Whitney U test. 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 produce 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. This means that 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]. We evaluate the impact of these three parameters in Appendix C.5. Unsurprisingly, larger values of ε reduce the number of detected events, but those that remain correspond to higher-amplitude degradations. Stricter p-value thresholds increase statistical confidence, but also bias detections toward user groups with more measurements, potentially missing events in less-measured regions. Finally, tightening the 80% requirement sharply decreases the number of events detected, effectively restricting anomalies to those that persist across the entire day. Overall, we flag a User Group as experiencing a latency *anomaly* when all three conditions are met: (i) the statistical tests yield a p-value below 0.05, (ii) the latency deviation exceeds the sensitivity threshold, and (iii) at least 80% of measurements have a higher latency than the baseline plus the sensitivity threshold.

4.1.3 Throughput

Detecting throughput anomalies is challenging because measurements fluctuate widely even under normal conditions, driven by factors like subscription plans [75], Wi-Fi quality [87], and home network setups. The pool of users running tests also changes over time, so samples often come from different populations, introducing distribution shifts unrelated to network performance.

These factors create three challenges: (i) throughput shows much greater variance than latency (Appendix B.7), complicating statistical detection ; (ii) measurement distributions are often multimodal, with distinct clusters caused by differences in access technology (e.g., fiber vs. DSL), subscription tiers, home network setups, and device capabilities, making single-baseline approaches unreliable [75]; and (iii) the composition of user groups shifts over time, altering mode distributions even without network changes. Anomalies appear either as sharp drops outside known modes or as significant shifts in the distribution of measurements across modes.

To address these challenges, we design a new statistical approach to detect throughput anomalies in the presence of multimodal distributions and shifting user populations. Our method computes the Wasserstein distance [94] between the baseline and observed throughput distributions to quantify changes in their shape and location. To determine if this shift is statistically significant, we perform a permutation test [36]: we repeatedly shuffle baseline and observed samples to generate a null distribution of Wasserstein distances, then compare the observed value against this null to compute a p-value. This combination captures subtle distribution shifts—such as changes in the proportion of users in different throughput tiers—that simple mean or median comparisons would miss (Appendix B.3). In addition to distributional shifts, as for latency, we also perform the Mann-Whitney U test to identify statistically significant median differences. We evaluate the impact of these parameters in Appendix C.5 and find that they exhibit trends consistent with those observed in the latency analysis. Overall, a User Group is flagged as experiencing an *anomaly* in throughput when the three following criteria are met: (i) the Wasserstein distance and the Mann-Whitney U tests yield p-values under 0.05, (ii) the median throughput difference exceeds a specified threshold, δ, to filter out minor variations, and (iii) at least 80% of measurements have a lower throughput than the median throughput minus the threshold the week earlier.

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, the Reverse Traceroute Sidecar [48, 78, 93] collects a reverse traceroute from the client back to the server, providing a bidirectional view of each path. In Appendix C.8, we quantify the fraction of reverse traceroute that are exploitable, which ranges between 52 to 66% comparable to prior results [93]. 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 B.2).

We represent each node as a $\langle \text{AS}, \text{metro} \rangle$ pair. This level of abstraction strikes a balance between fidelity and scala-

bility: 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 nodes or links in a network are responsible for observed problems using only end-to-end measurements. This problem is often underspecified: 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 [31, 35, 56]. Events can also occur at multiple levels of granularity—from peering links to fibers, facilities, metro, or entire ASes [92]. Mapping across these levels adds complexity because failures at one granularity can mask or cascade into another.

Traditional network tomography methods rely on assumptions that do not hold at Internet scale. Many require dense measurement coverage [35] or dedicated probing infrastructure [26], while others use linear inference models that are sensitive to noise and do not scale to large, sparse datasets [38]. In our setting, measurements are sparse, paths are asymmetric, and vantage points are limited, making these approaches impractical. Prior studies have also shown that general-purpose inference algorithms are computationally expensive [17, 18, 29, 66], making it infeasible to run them across the hundreds of thousands of $\langle \text{AS}, \text{metro} \rangle$ pairs in our topology. These constraints motivate the need for a lightweight, scalable methodology tailored for Internet-wide measurement.

Our framework localizes disruptions at multiple levels of granularity, from AS-wide outages to specific links, using two complementary techniques: *Temporal Tomography* highlights changes in the fraction of anomalous paths traversing a given network entity before and during an event, surfacing entities that may have become newly involved or avoided. In contrast, *correlation tomography* isolates entities that consistently appear on anomalous paths, even in the absence of visible routing changes. This dual approach is necessary because not all anomalies stem from rerouting—congestion may degrade performance along stable paths, while load-balancing changes may shift routes without causing degradation. By combining both perspectives, our system distinguishes between structural disruptions caused by routing changes and degradations caused by congestion along stable paths. Our algorithm is inspired by 007 [5], but whereas 007 considers highly-structured and symmetric data center topologies where a voting mechanism can accurately pinpoint failures, applying such voting to a measured view of the Internet's heterogeneous topology risks introducing bias. We refer to the paths associated with these speed-tests as *anomalous paths*. Correlation tomography complements this by isolating the network entities most likely responsible for the observed anomalies. We further extend this framework to operate on a hierarchical model that attributes responsibility beginning with the broadest level and progressively refining to more granular levels.

4.3.1 Temporal Tomography

To identify links most likely responsible for an event at time t_{event} (day granularity), we compare each link's anomaly rate during the event against its baseline. In particular, 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* here 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 $A \to B$ 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 $A \to B$ 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 clear 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 develop a method that locates which network is most likely responsible for an event by building a topology that includes

all $\langle \text{AS}, \text{metro} \rangle$ nodes present in any forward or reverse traceroute from user groups experiencing the event on the target day and using this topology to localize the problems.

Naive strategies, such as counting anomalies per link, overrepresent large transit providers because they naturally appear on many paths. Using only the fraction of anomalous paths per link is also unreliable, as sparsely observed links can appear anomalous by chance. To address this, we use two complementary metrics: (i) the fraction of all anomalies that traverse a link, assessing whether a significant share of anomalies could be attributed to it, and (ii) the fraction of its paths that are anomalous, reducing false positives from sparse data.

We identify likely culprits using an iterative filtering process (Alg.1 in Appendix B.4). At each iteration, we select the link with the highest anomaly ratio (anomalous vs. total paths) that exceeds a sensitivity threshold ε. All anomalous source-destination pairs that traverse this link are then considered "explained" and removed from the pool of remaining anomalies, we repeat this process until no links meet the criteria. For example, consider a link $A \to B$ that appears in 8% of anomalous paths but only 3% of all paths and such that the ratio 8/3 is the highest of the dataset. Given that the fraction of anomalous paths is significantly higher, and assuming that 50% ($> \varepsilon$) of all paths crossing $A \to B$ are anomalous, this edge is flagged as a candidate for originating the disruption affecting its downstream user groups. Once a link is identified as a culprit, all anomalies it explains are removed from further iterations. This follows the intuition that a single event is unlikely to originate from multiple unrelated parts of the network simultaneously, an assumption supported by prior work [5, 47]. The process continues iteratively, selecting the next most likely culprit until either all anomalies are explained or no remaining edge meets the selection criteria.

After identifying all potential problematic links, we aggregate them by $\langle \text{AS}, \text{metro} \rangle$ nodes and apply thresholds to identify whether the anomaly happens at this granularity. We perform similar aggregation on each hyperedge across AS, metro, facility, or IXP boundaries to detect broader failures. Finally, our algorithm assigns explanations to each event, prioritizing them first at the metro level, then facility, then IXP, then AS, $\langle \text{AS}, \text{metro} \rangle$, and only narrowing down to specific links if necessary.

Combining the two approaches. Each day, we run Temporal Tomography to flag $\langle \text{AS}, \text{metro} \rangle$ links, nodes, ASes, and metros with the largest Δ_ℓ changes, and Correlation Tomography to identify entities most strongly associated with anomalies.

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 [35]. Resolving these ambiguities requires additional active measurements, but 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 optimization central to our design.

We use a post-hoc measurement planning step to direct new measurements where they have the most diagnostic value. Because on-demand NDT tests would overload client connections, we focus 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. To guide this process, we build on metAScritic [83], 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. This process, along with the scoring and selection algorithm behind it, is described in more detail in Appendix B.5. Because measurement capacity is 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. Our default eviction rule is simple: 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

Overview: Validating HERMES is challenging due to the lack of ground-truth data on performance degradations and their sources—precisely the gap HERMES aims to address. We evaluate HERMES against multiple independent datasets, using agreement as evidence of correctness while recognizing that no single dataset provides complete visibility. We focus on three questions: *Precision*: Are disruptions flagged by HERMES consistent with well-instrumented systems? *Recall*: How many documented disruptions does HERMES detect? *Coverage*: Does

HERMES have sufficient geographic and network reach to serve as a global observatory? We count a true positive when a HERMES event matches an external report in time (same day) and location (same $\langle \text{AS}, \text{metro} \rangle$ or containing region); a false positive when HERMES flags an event on a well-observed path where the other dataset shows none; a false negative when the other dataset reports an event on a network we measure yet HERMES misses it; and a true negative when both agree no issues exist. 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 (Appendix D.1), enabling operators and researchers to inspect results.

HERMES achieves high precision despite relying solely on public data. Compared to a reimplementation of BlameIT in AnonCDN (§5.1), 91.4% of events flagged by HERMES are also identified by BlameIT, establishing a lower bound on precision. Among the events detected by both systems, HERMES correctly identifies the same responsible segment for 94.5% of cases. For recall, we compare HERMES's detections against ISP outage reports, operator-verified tickets, mailing lists, and user-reported incidents (§5.2), finding recall rates between 85.1% and 64.7%. Combining all the datasets, only 5.2% of the events were detected by the existing IODA and CloudFlare Radar observatories [24, 42], demonstrated that HERMES complements existing systems. We evaluate HERMES's coverage across logical and geographic dimensions, showing that its measurements span ASes serving over 95% of the global Internet population and include sufficient data in the world's largest metro areas to reliably detect network failures (§5.3). We demonstrate the added value of disambiguating measurements in Appendix C.6, where targeted traceroutes reduce ambiguity by 47% on average, fully resolve 31% of cases, and outperform random probing by resolving 75% more ambiguity sets. Finally, we show that reverse path visibility is crucial for diagnosing the network events, as routing changes and congestion frequently occur asymmetrically. Our analysis shows that 50.5% of links that were the sources of events require bidirectional path data for correct attribution, and reverse path rerouting explains a larger fraction of latency spikes during events compared to the forward path (§5.4).

5.1 Comparing to a CDN's Monitoring System

To evaluate HERMES's precision, we compare it to BlameIT, a peer-reviewed system for localizing faults in Microsoft's network [47]. We re-implemented BlameIT within AnonCDN using its internal telemetry data, providing a strong reference point for validation. BlameIT organizes measurements into quartets defined by client /24, CDN site, device type, and 5-minute time windows. BlameIT groups measurements into quartets defined by client /24, CDN site, device type, and 5-minute windows. It then classifies each quartet as good or bad using latency thresholds and attributes problems to one of three coarse path segments: client, middle, or CDN. BlameIT also uses active traceroute probes to refine this attribution (in particular when the problem is in the middle of the segment). Since we were not able to replicate the traceroute portion of BlameIT in AnonCDN, we assume that a user group experiencing an event categorized as occurring in the "middle segment" in BlameIT is equivalent to HERMES identifying a hop in the middle of the path as the source for the event.

Results: We cannot share exact numbers from AnonCDN but instead provide order-of-magnitude estimates in Table 2. A large portion of the events detected by HERMES (91.4%) are also observed by AnonCDN, demonstrating that HERMES is not raising false alarms because of its reliance on relatively sparse speedtests, as the events are also visible in the much denser CDN dataset. This is a *lower bound*, as the remaining 9% could theoretically represent events occurring on paths not traversed by traffic between AnonCDN and its users. Although our analysis considers these false positives, they are actually events invisible to AnonCDN's BlameIT vantage points. For shared events, HERMES matches BlameIT's segment attribution 94.5% of the time, demonstrating precision comparable to cloud-scale telemetry despite using only public data.

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.[2] 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). Publicly reported disruptions are sourced from the NANOG and Outage mailing lists [65, 67], long-standing forums where experts share and vet outage reports [7], and from Reddit, which provides large-scale but less reliable user-reported accounts of connectivity issues. We provide full details of how these datasets were collected, filtered, and validated in Appendix C.1.[3] By combining operator-curated data (high accuracy, low coverage) with public reports (broad coverage, variable reliability), we balance scale and confidence, ensuring validation captures both well-documented outages and less certain, user-impacting disruptions that traditional monitoring overlooks.

Table 2 summarizes HERMES's detection and classification performance across validation sources, listed in order of decreasing reliability. Among high-confidence sources, HERMES

[2] We attempted to replicate the methodology of prior work that identified outages using Google Trends [50]. However, changes to the Google Trends API have rendered the paper's tool unusable and, more broadly, have significantly complicated the process of automatically crawling Google Trends compared to the past [62].

[3] A Reddit post maps to a HERMES event if several users report degradation in the same AS/metro in the same day (more details can be found in Appendix C.2). A single Reddit thread may correspond to multiple HERMES events if users in different regions and ASes are affected.

Table 2: HERMES's recall and precision across various sources.

Source	Total Events	Recall (%)	Precision (%)
AnonCDN's BlameIT	≈ 100Ks[4]	-	86.5%
ISP Outages	27	85.1%	-
AnonCDN's Tickets	≈ 10s	77.9%	-
Mailing Lists	42	73.8%	-
Reddit (Confirmed)	213	64.7%	-
Reddit	3207	57.2%	-

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 identified by the LLM, HERMES captured 57.2%. This represents 12× more issues detected than existing public observatories such as Cloudflare Radar and IODA (see Appendix C.2 for details on dataset processing).

Operators' outage pages—authoritative confirmations: ISP-operated outage pages provide official acknowledgments of service disruptions [25, 39, 73, 86]. Since these pages often display only current events, we use the Internet Archive's Wayback Machine to retrieve all historical snapshots from August 2024. We identified and archived these pages through manual searches for 4 networks to cross-reference them with HERMES's detections. The archived data primarily covers outages involving transit networks, making this dataset particularly valuable to evaluating whether HERMES correctly attributes the source of an event.[5]

Result: HERMES successfully detected 23 out of 27 documented outages (85.1%). In 2 cases, it incorrectly attributed the source of the failure; in 2 others, it detected no event within the affected subgroups.

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 incidents where faults occurred outside AnonCDN's infrastructure or direct peering links, as we are primarily interested in events impacting public Internet paths. Specifically, we identified tens of cases where operators determined that the issue was caused by an AS along the path between the client's ISP and AnonCDN (excluded). These events were often characterized by reduced goodput, increased latency, and loss. 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 78.1% of the operator-validated events. Most instances when HERMES missed an event, it was because of insufficient measurements. Otherwise, it localized 4 incorrectly and misclassified 3 as non-anomalous behavior.

5.3 Coverage of Internet's Users

Reliable detection of Internet-wide events depends on having *sufficient measurement density* within each network and location. As a reminder, HERMES only considers a metro-AS group if it produces at least 25 tests from 5 or more unique IPs over a week, which provides enough statistical power for anomaly detection without being skewed by any single user. To evaluate HERMES's coverage, we examine two dimensions: logical and geographic. Logical coverage evaluates whether HERMES can detect failures in networks that serve large user populations. To approximate this, we combine two complementary datasets: APNIC's user population estimates [4], which provide accurate per-AS user counts at the country level [81], and IPInfo's prefix-level geolocation data [44], which identifies metros where ASes operate and flags prefixes likely to host human users.[6] APNIC supplies scale—how many users are behind each AS—while IPInfo provides localization—where those users likely reside. We distribute each AS's country-level user population across its observed metro footprints in IPInfo, weighted by metro population, under the assumption that Internet users in an AS track general population distribution across its metros. This combined approach is necessary because neither dataset alone is sufficient: APNIC cannot break down users below the country level, while IPInfo does not estimate the number of users behind each prefix or AS. Using them together allows us to approximate metro-level coverage, though inaccuracies can arise from errors in inferred AS footprints or proportional allocation. Finally, we note that APNIC's data is derived from Google Ads measurements, so countries with limited Google presence (e.g., Russia [81, 98]) may be under-represented. These values should therefore be treated as approximate rather than exact.

Figure 4 illustrates the fraction of the estimated Internet user population in each country for which HERMES has sufficient measurements (*i.e.,* at least 25 tests from 5 unique sources). HERMES covers between 40% and 90% of Internet users depending on the region, with the strongest coverage in Europe and North America and the weakest in parts of Africa. Coverage in the US appears artificially low because large corporate networks (e.g., CDNs and cloud providers) are frequently mis-geolocated by MaxMind and IPInfo to US locations. These prefixes do not actually represent residential

[5] Access network operators typically provide outage information through dashboards that require users to input their addresses or their account numbers.

[6] From a private discussion with IPInfo, these flags are set when the data come from mobile devices or direct communication with operators.

users, but their misclassification inflates the denominator of "Internet users" in the US, making it seem as though a larger fraction of the population is uncaptured by HERMES than is truly the case. In Brazil, APNIC reports a highly fragmented ecosystem of small access ISPs, many of which contribute only a handful of measurements—fewer than our sufficiency threshold—leading to under-coverage despite a large absolute user base [81]. Appendix C.4 expands this analysis to additional dimensions of coverage. In Appendix C.4.1, we show that in every major metro, at least one ISP has enough measurements to support continuous monitoring at all times. In Appendix C.4.2, we evaluate coverage at the infrastructure level by measuring the fraction of IXPs, ASes, and PoPs traversed by our tests, illustrating the breadth of network events that HERMES can detect.

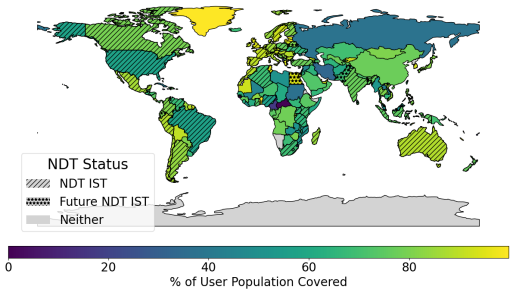


Figure 4: User population coverage by HERMES, with shading indicating the percentage of the population covered per country. Hatching denotes deployment status: diagonal lines for countries currently served by Google's Internet Speed Test in Search (*NDT IST*), stars for countries with planned deployment (*Future NDT IST*), and no hatching for countries not covered.

5.4 Importance of Bidirectional Paths

To demonstrate the necessity of measuring both the server-to-user (forward) and user-to-server (reverse) paths when diagnosing network performance issues, we examine three questions: (i) How often do performance changes correlate with shifts in the geographic or logical paths in either direction? (ii) How much of the total observed latency can be accounted for by analyzing the forward path alone, the reverse path alone, or both together? (iii) How would the results of our tomography differ if we omitted information from either direction?

For (i), we evaluate whether latency and throughput degradations are associated with routing changes by analyzing shifts in the AS-level and geographic paths in both directions. We compare the paths of measurements with anomalous performance to other observed paths in the preceding day. If a change is detected in the AS-level or geographic dimension for either direction, we classify the event as linked to 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 there is no reason to believe that this subset is not representative of the broader dataset and the results we expect from HERMES as MLab 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. Additionally, not all performance issues stem from routing changes as discussed in Section 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 Path Changes	3.1
Forward AS Path	19.8	Only Forward Path Changes	1.5
Reverse Geographic Path	26.6	Both Change	25.5
Forward Geographic Path	24.1	At Least One Changes	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 on events. For each path, we geolocate the IP addresses to calculate the path length, then convert to latency based on the speed of light in fiber [88] to estimate a lower bound on latency (ignoring geolocation errors, missing hops, and circuitous routes between adjacent hops). When using only the forward path, we assume the reverse path follows the great-circle route, and vice versa. 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, this pattern holds at scale, *i.e.,* the reverse path provides, on average, more insight than the forward path in explaining latency on paths during events. This analysis provides a lower bound on the contribution of routing to the observed latency. Even when considering both forward and reverse paths, the best geographic-based approximation of latency we can construct, a significant portion of latency remains unexplained. This phenomenon can be attributed to the limitations of traceroutes probing (*e.g.,* layer-2 tunnels, unresponsive hops, processing delays).

For (iii), we study how tomography results change when omitting forward or reverse path information by analyzing each direction's role in identifying event-causing links and the fraction of anomalous and regular paths that cross them. We compare three scenarios: detecting anomalous links (a) using only the forward path, (b) only the reverse path, and (c) both together. We then measure the percentage of links that could be attributed as the source of an event using a single direction versus those that required bidirectional visibility for detection. Up to 24.6% of links were identifiable using only the forward path, while 24.9% were detected using only the reverse path. However, 50.5% of links required information from both directions to be classified as sources.

These results highlight that forward *and* reverse paths are needed to accurately identify the source of performance issues.

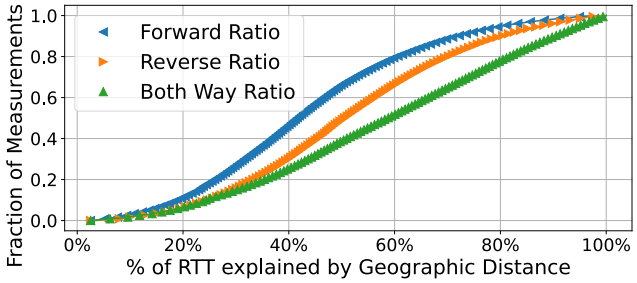


Figure 5: Distribution of the fraction of RTT explained by geographic distance for forward, reverse and both-way paths. In particular, the reverse path is most often informative than the forward path to explain the latency.

Hence reverse path visibility is *essential*. Any system that only relies on one direction might misattribute events and miss their source. In Appendix D.2, 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.

6 What can we see with HERMES?

Over a period of 5 month, HERMES has processed approximately 4 billion speed tests, identifying 65K events across 16K cities in 166 countries, impacting clients in 9,710 ASes. To ensure that HERMES's insights are both accessible and actionable, we fully automated the system and developed a detailed dashboard. We plan to share this dashboard with the community upon acceptance (a screenshot is provided in Appendix D.1). In addition to these case studies we present in this section, Appendix D.2 shows that 72% of paths have a more circuitous reverse route, often twice as long, around periods of anomalies, and Appendix D.3 discusses how HERMES can be used to infer points of congestions at interconnections.

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

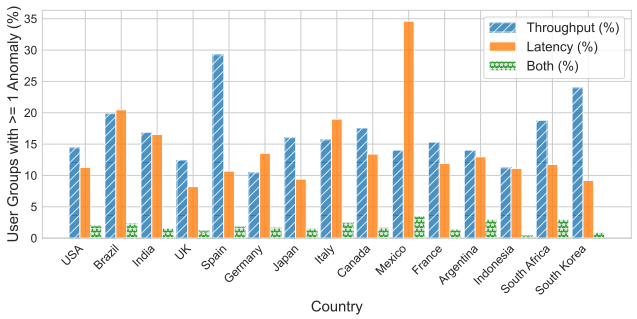


Figure 6: Percent of user groups with episodes of latency or throughput degradation in the 15 countries with the highest anomaly counts (ranked by the total anomalies over a 30-day period).

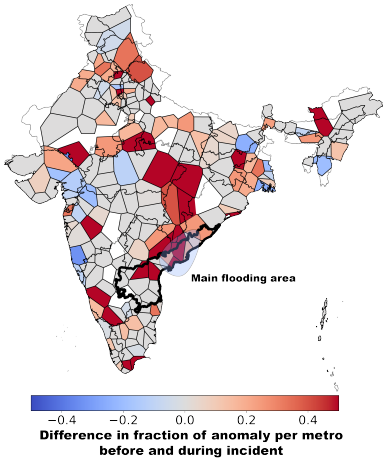


Figure 7: Difference in the fraction of user groups with detected anomalies across metro areas in India before (August 24-31, 2024) and during (September 1-8, 2024) the flooding in Andhra Pradesh.

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, 84, 89], events like protests, cable cuts, or extreme weather can degrade the performance of PoPs and cables, leading to congestion, throttled connectivity, 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.

As an example, we use HERMES to analyze how anomalies vary across ASes in metros affected by these events. Our primary case study examines the severe flooding in Andhra Pradesh, India, in September 2024 [90]. Figure 7 compares the difference in the fraction of ASes in each metro with anomalies before and during the flooding. We observe that many of the areas close to the main flooding area experienced an increase (above 0.4) in their fraction of anomalies. We also find that network disruptions extend beyond the directly affected region, impacting surrounding areas, likely because of complex interdependencies in network routing and infrastructure. Appendix D.4 presents case studies and similar figures for the Valencia flooding [6], Japan's Typhoon Shanshan[95], the Baltic Sea cable cut [8], and Hurricane Helene [34].

7 Conclusion

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.

References

[1] Peeringdb, 2025. PeeringDB is a freely available, user-maintained database of networks and interconnection data to support the global Internet ecosystem.

[2] Scott Anderson, Loqman Salamatian, Zachary S Bischof, Alberto Dainotti, and Paul Barford. igdb: connecting the physical and logical layers of the internet. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 433–448, 2022.

[3] Anonymous. Is cogent down in chicago? https://old.reddit.com/r/sysadmin/comments/1e41e7y/is_cogent_down_in_chicago/, 2013. Accessed: 2024-12-02.

[4] APNIC. Customers per as measurements – visible asns: Customer populations (est.). https://stats.labs.apnic.net/aspop/, n.d. Accessed: 2025-09-17.

[5] Behnaz Arzani, Selim Ciraci, Luiz Chamon, Yibo Zhu, Hongqiang Harry Liu, Jitu Padhye, Boon Thau Loo, and Geoff Outhred. 007: Democratically finding the cause of packet drops. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 419–435, 2018.

[6] Associated Press. Massive flooding in valencia due to torrential rains. https://www.apnews.com/articles/valencia-2024-floods, 2024. Accessed: 2024-10-29.

[7] Ritwik Banerjee, Abbas Razaghpanah, Luis Chiang, Akassh Mishra, Vyas Sekar, Yejin Choi, and Phillipa Gill. Internet outages, the eyewitness accounts: Analysis of the outages mailing list. In *International Conference on Passive and Active Network Measurement*, pages 206–219. Springer, 2015.

[8] BBC News. Severe disruptions due to baltic sea cable cuts. https://www.bbc.com/news/baltic-sea-cable-cuts-2024, 2024. Accessed: 2024-10-17.

[9] Karyn Benson, Alberto Dainotti, Kc Claffy, Alex C Snoeren, and Michael Kallitsis. Leveraging internet background radiation for opportunistic network analysis. In *Proceedings of the 2015 Internet Measurement Conference*, pages 423–436, 2015.

[10] David Blackwell and JL Hodges Jr. Design for the control of selection bias. *The Annals of mathematical statistics*, 28(2):449–460, 1957.

[11] CAIDA. The caida internet topology data kit (itdk). Dataset, 2023. Accessed: 2024-01-30.

[12] CAIDA. As rank - autonomous system rankings. https://asrank.caida.org/, 2024. Accessed: 2024-12-09.

[13] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the performance of an anycast cdn. In *Proceedings of the 2015 Internet Measurement Conference*, IMC '15, page 531–537, New York, NY, USA, 2015. Association for Computing Machinery.

[14] Neal Cardwell, Ian Swett, and Joseph Beshay. BBR Congestion Control. Internet-Draft draft-ietf-ccwg-bbr-01, Internet Engineering Task Force. Work in Progress.

[15] Esteban Carisimo, Rashna Kumar, Caleb J. Wang, Santiago Klein, and Fabián E. Bustamante. Ten years of the venezuelan crisis - an internet perspective. In *Proceedings of the ACM SIGCOMM 2024 Conference*, ACM SIGCOMM '24, page 521–539, New York, NY, USA, 2024. Association for Computing Machinery.

[16] Esteban Carisimo, Ricky K. P. Mok, David D. Clark, and K. C. Claffy. Jitterbug: A new framework for jitter-based congestion inference. In *Passive and Active Measurement: 23rd International Conference, PAM 2022, Virtual Event, March 28–30, 2022, Proceedings*, page 155–179, Berlin, Heidelberg, 2022. Springer-Verlag.

[17] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. 2004.

[18] Yan Chen, David Bindel, Hanhee Song, and Randy H. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, page 55–66, New York, NY, USA, 2004. Association for Computing Machinery.

[19] David R Choffnes, Fabián E Bustamante, and Zihui Ge. Crowdsourcing service-level network event monitoring. In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 387–398, 2010.

[20] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[21] David D Clark and Sara Wedeman. Measurement, meaning and purpose: Exploring the m-lab ndt dataset. In *TPRC49: The 49th Research Conference on Communication, Information and Internet Policy*, 2021.

[22] Cloudflare. Cloudflare radar: United states traffic on july 4th, 2024. Accessed: 2024-07-16.

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

[24] Inc. Cloudflare. Cloudflare radar, 2024. Accessed: 2024-12-04.

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

[26] Ítalo Cunha, Renata Teixeira, Nick Feamster, and Christophe Diot. Measurement methods for fast and accurate blackhole identification with binary tomography. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, page 254–266, New York, NY, USA, 2009. Association for Computing Machinery.

[27] Omar Darwich, Hugo Rimlinger, Milo Dreyfus, Matthieu Gouel, and Kevin Vermeulen. Replication: Towards a publicly available internet scale ip geolocation dataset. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 1–15, 2023.

[28] 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. Inferring persistent interdomain congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 1–15, New York, NY, USA, 2018. Association for Computing Machinery.

[29] Amogh Dhamdhere, Renata Teixeira, Constantine Dovrolis, and Christophe Diot. Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data. In *Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, 2007.

[30] Downdetector. Downdetector: Real-time problem and outage monitoring, 2024. Accessed: 2024-05-12.

[31] Nick Duffield. Network tomography of binary network performance characteristics. *IEEE Transactions on Information Theory*, 52(12):5373–5388, 2006.

[32] Mah-Rukh Fida, Andres F Ocampo, and Ahmed Elmokashfi. Measuring and localising congestion in mobile broadband networks. *IEEE Transactions on Network and Service Management*, 19(1):366–380, 2021.

[33] Romain Fontugne, Cristel Pelsser, Emile Aben, and Randy Bush. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. In *Proceedings of the 2017 Internet Measurement Conference*, pages 15–28, 2017.

[34] Fox Weather. Hurricane helene: Catastrophic flooding and widespread destruction. https://www.foxweather.com/weather-news/top-weather-stories-2024, 2024. Accessed: 2024-10-09.

[35] Denisa Ghita, Can Karakus, Katerina Argyraki, and Patrick Thiran. Shifting network tomography toward a practical goal. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[36] Phillip Good. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media, 2013.

[37] Jiyao Hu, Zhenyu Zhou, Xiaowei Yang, Jacob Malone, and Jonathan W. Williams. CableMon: Improving the reliability of cable broadband networks via proactive network maintenance. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI '20)*, pages 619–632, Santa Clara, CA, February 2020. USENIX Association.

[38] Yiyi Huang, Nick Feamster, and Renata Teixeira. Practical issues with using network tomography for fault diagnosis. *SIGCOMM Comput. Commun. Rev.*, 38(5):53–58, September 2008.

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

[40] Hurricane Electric. Bgp toolkit. https://bgp.he.net/, 2024. Accessed: 2024-12-09.

[41] Geoff Huston. Analyzing the internet's bgp behavior. https://www.ece.ucf.edu/~yuksem/teaching/ip/reading/huston-bgp.pdf, 2001. Accessed: 2025-01-17.

[42] Georgia Institute of Technology Internet Intelligence Research Lab. Internet outage detection and analysis (ioda), 2024. Accessed: 2024-12-04.

[43] IODA. Ioda: Asn 174 (cogent) monitoring data, 2024. Accessed: 2024-07-16.

[44] IPInfo.io. Ip geolocation api and database, 2025. Accessed: 2025-01-24.

[45] IsItDownRightNow. Is it down right now? - website down or not?, 2024. Accessed: 2024-05-12.

[46] Umar Javed, Italo Cunha, David Choffnes, Ethan Katz-Bassett, Thomas Anderson, and Arvind Krishnamurthy. Poiroot: Investigating the root cause of interdomain path changes. *ACM SIGCOMM Computer Communication Review*, 43(4):183–194, 2013.

[47] Yuchen Jin, Sundararajan Renganathan, Ganesh Ananthanarayanan, Junchen Jiang, Venkata N Padmanabhan, Manuel Schroder, Matt Calder, and Arvind Krishnamurthy. Zooming in on wide-area latencies to a global cloud provider. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 104–116, 2019.

[48] Ethan Katz-Bassett, Harsha V. Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter van Wesep, Thomas Anderson, and Arvind Krishnamurthy. Reverse traceroute. In *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 10)*, San Jose, CA, April 2010. USENIX Association.

[49] Ethan Katz-Bassett, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, David Wetherall, and Thomas Anderson. Studying black holes in the internet with hubble. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, page 247–262, USA, 2008. USENIX Association.

[50] Ege Cem Kirci, Martin Vahlensieck, and Laurent Vanbever. "is my internet down?": Sifting through user-affecting outages with google trends. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 290–297, New York, NY, USA, 2022. Association for Computing Machinery.

[51] Thomas Koch, Shuyue Yu, Sharad Agarwal, Ethan Katz-Bassett, and Ryan Beckett. Painter: Ingress traffic engineering and routing for enterprise cloud networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 360–377, 2023.

[52] Rupa Krishnan, Harsha V Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving beyond end-to-end path information to optimize cdn performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 190–201, 2009.

[53] Measurement Lab. Isp interconnection and its impact on consumer internet performance. Technical report, Measurement Lab, 2014. Accessed: 2025-01-24.

[54] Matthew Luckie, Amogh Dhamdhere, David Clark, Bradley Huffaker, and KC Claffy. Challenges in inferring internet interdomain congestion. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 15–22, 2014.

[55] Matthew Luckie, Bradley Huffaker, Alexander Marder, Zachary Bischof, Marianne Fletcher, and K Claffy. Learning to extract geographic information from internet router hostnames. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '21, page 440–453, New York, NY, USA, 2021. Association for Computing Machinery.

[56] Liang Ma, Ting He, Ananthram Swami, Don Towsley, Kin K Leung, and Jessica Lowe. Node failure localization via network tomography. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 195–208, 2014.

[57] Kyle MacMillan, Tarun Mangla, James Saxon, Nicole P. Marwell, and Nick Feamster. A comparative analysis of ookla speedtest and measurement labs network diagnostic test (ndt7). *Proc. ACM Meas. Anal. Comput. Syst.*, 7(1), March 2023.

[58] Harsha Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI 06)*, Seattle, WA, November 2006. USENIX Association.

[59] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, kc claffy, and Jonathan M. Smith. Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, page 56–69, New York, NY, USA, 2018. Association for Computing Machinery.

[60] Measurement Lab. MSAK (Measurement Swiss-Army Knife). https://www.measurementlab.net/tests/msak/, 2025. Configurable WebSocket-based multi-stream throughput test and UDP-based latency test; publicly available data via Google Cloud Storage and BigQuery.

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

[62] General Mills. Issue #561 - api changes breaking pytrends usage. https://github.com/GeneralMills/pytrends/issues/561, 2023. Accessed: 2025-01-17.

[63] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. A multifaceted look at starlink performance. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2723–2734, New York, NY, USA, 2024. Association for Computing Machinery.

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

[65] Puck Nether Net. Outages mailing list, 2024. Accessed: 2024-12-04.

[66] H. X. Nguyen and P. Thiran. The boolean solution to the congested ip link location problem: Theory and practice. In *IEEE INFOCOM 2007 - 26th IEEE International*

Conference on Computer Communications, pages 2117–2125, 2007.

[67] North American Network Operators' Group (NANOG). Nanog mailing list. https://mailman.nanog.org/mailman/listinfo/nanog, 2025. Accessed: 2025-01-17.

[68] Ookla. Ookla open data. https://www.ookla.com/open-data, 2025. Accessed: 2025-01-27.

[69] Open Technology Fund. OONI: Open Observatory of Network Interference, 2024. Accessed: 2024-12-09.

[70] OpenAI. Chatgpt-4 api. https://platform.openai.com/docs/, 2024. Accessed: 2024-12-04.

[71] Packet Clearing House. Internet exchange points directory. https://www.pch.net/services/internet_exchange_points, 2024. Accessed: 2024-12-09.

[72] Ramakrishna Padmanabhan, Aaron Schulman, Dave Levin, and Neil Spring. Residential links under the weather. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 145–158, 2019.

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

[74] Udit Paul, Jiamo Liu, David Farias-llerenas, Vivek Adarsh, Arpit Gupta, and Elizabeth Belding. Characterizing internet access and quality inequities in california m-lab measurements. In *Proceedings of the 5th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*, COMPASS '22, page 257–265, New York, NY, USA, 2022. Association for Computing Machinery.

[75] Udit Paul, Jiamo Liu, Mengyang Gu, Arpit Gupta, and Elizabeth Belding. The importance of contextualization of crowdsourced active speed test measurements. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 274–289, 2022.

[76] Photon Shift Project. Photon reddit data download tool, 2024. Accessed: 2025-01-24.

[77] Lin Quan, John Heidemann, and Yuri Pradkin. Trinocular: Understanding internet reliability through adaptive probing. *ACM SIGCOMM Computer Communication Review*, 43(4):255–266, 2013.

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

[79] RIPE NCC. Ripe ipmap - infrastructure geolocation. https://labs.ripe.net/tools/ripe-ipmap/ripe-ipmap/, 2024. Accessed: 2024-12-09.

[80] Route Views Project. Route views project - bgp data. https://www.routeviews.org/routeviews/, 2024. Accessed: 2024-12-09.

[81] Loqman Salamatian, Calvin Ardi, Vasileios Giotsas, Matt Calder, Ethan Katz-Bassett, and Todd Arnold. What's in the dataset? unboxing the apnic per as user population dataset. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 165–182, New York, NY, USA, 2024. Association for Computing Machinery.

[82] Loqman Salamatian and Phillipa Gill. How M-Lab Determines User Location and Selects Servers. Measurement Lab blog, May 2025. https://www.measurementlab.net/blog/improving-m-lab-geolocation/.

[83] Loqman Salamatian, Kevin Vermeulen, Italo Cunha, Vasilis Giotsas, and Ethan Katz-Bassett. metascritic: Reframing as-level topology discovery as a recommendation system. In *ACM Internet Measurement Conference (IMC'24)*, 2024.

[84] Aaron Schulman and Neil Spring. Pingin' in the rain. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, page 19–28, New York, NY, USA, 2011. Association for Computing Machinery.

[85] Amazon Web Services. Amazon cloudwatch weather map. https://aws.amazon.com/cloudwatch/, 2024. Acessed: 2024-12-03.

[86] SFR. SFR Network Status Page. https://www.sfr.fr/media/export-arcep/siteshorsservices.csv. Accessed: 2025-01-22.

[87] Ranya Sharma, Nick Feamster, and Marc Richardson. A longitudinal study of the prevalence of wifi bottlenecks in home access networks. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 44–50, New York, NY, USA, 2024. Association for Computing Machinery.

[88] Sikich LLP. Networking at the speed of light: Understanding fiber optics. Sikich Blog, 2024. Accessed: 2024-12-09.

[89] Xiao Song, Guillermo Baltra, and John Heidemann. Inferring changes in daily human activity from internet response. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 627–644, 2023.

[90] Sphere India. AP Flood Situation Report: Sitrep-1, October 2024. Accessed: 2025-01-23.

[91] Srikanth Sundaresan, Xiaohong Deng, Yun Feng, Danny Lee, and Amogh Dhamdhere. Challenges in inferring

internet congestion using throughput measurements. In *Proceedings of the 2017 Internet Measurement Conference*, pages 43–56, 2017.

[92] William Sussman, Emily Marx, Venkat Arun, Akshay Narayan, Mohammad Alizadeh, Hari Balakrishnan, Aurojit Panda, and Scott Shenker. The case for an internet primitive for fault localization. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, pages 160–166, 2022.

[93] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, David Choffnes, and Ethan Katz-Bassett. Internet scale reverse traceroute. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 694–715, New York, NY, USA, 2022. Association for Computing Machinery.

[94] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[95] Wikipedia Contributors. Typhoon shanshan: Flooding and widespread damage. https://en.wikipedia.org/wiki/Typhoon_Shanshan_(2024), 2024. Accessed: 2024-08-28.

[96] WorldPop. Worldpop gridded population data. https://www.worldpop.org/, 2025. Population density estimates at 10 km resolution. Available at https://www.worldpop.org/.

[97] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. PlanetSeer: Internet path failure monitoring and characterization in Wide-Area services. In *6th Symposium on Operating Systems Design & Implementation (OSDI 04)*, San Francisco, CA, December 2004. USENIX Association.

[98] Zesen Zhang, Jiting Shen, and Ricky K. P. Mok. Empirical characterization of ookla's speed test platform: Analyzing server deployment, policy impact, and user coverage. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0630–0636, 2024.

[99] Jiangchen Zhu, Kevin Vermeulen, Italo Cunha, Ethan Katz-Bassett, and Matt Calder. The best of both worlds: high availability cdn routing without compromising control. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 655–663, New York, NY, USA, 2022. Association for Computing Machinery.

A Ethics

This study uses speed-test data to investigate whether specific user groups experience outages. While our analysis relies on anonymized, aggregate data and excludes any personally identifiable information (PII), it is essential to address the ethical considerations inherent in this work.

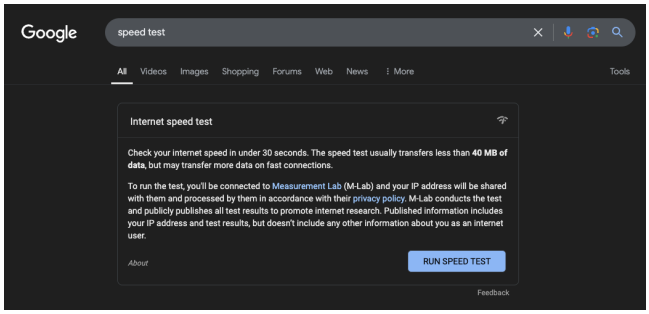


Figure 8: M-Lab NDT served via the Google One-Box plugin, which appears as the top option when searching for Internet speed-related queries on Google Search.

First, speed-test data originates from users who may not be fully aware of its use beyond testing their internet connection. Even though the data is anonymized and aggregated, and our use of it falls under the stated terms of use of the test tool, researchers bear a responsibility to ensure that its use respects user expectations.

Second, this work focuses on event detection at the metro-ASN level and attempts to attribute the source of these network issues. While we have carefully evaluated our methodology and adopted conservative approaches when attribution is uncertain, there is a possibility that our technique could misattribute the source of an event, assigning blame incorrectly. Such errors could have unintended consequences, including reputational harm to networks. For instance, highlighting persistent issues in certain regions or networks might inadvertently draw attention to areas with poor *physical* connectivity, potentially casting operators in a negative light without fully considering the broader context in which they operate. We strive to present our results constructively, focusing on fostering improvements rather than assigning blame.

To address these concerns, we (i) share results responsibly and collaboratively, opening them to all, including network operators, policymakers, and the research community to promote transparency and constructive action via several dashboards, and (ii) strictly adhere to the ethical guidelines and data-sharing policies established by M-Lab.

While we take these precautions, we recognize that ethical challenges cannot be entirely eliminated. We welcome open dialogue and critical evaluation of our methods to ensure alignment with broader principles of equity, fairness, and responsible research.

B Methodology Details

B.1 Cleaning geolocation data

HERMES groups measurements at the *metro* level, so accurate client geolocation is critical.

Combining server-selection and dataset geolocation to improve geolocation. M-Lab uses two independent systems to geolocate clients, one for server selection at test time and another for dataset annotation [82]. 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. Later, in the public BigQuery dataset, client IPs are annotated with MaxMind's GeoLite2 database, which is updated daily. While MaxMind achieves high country-level accuracy (99.8%), its city-level accuracy is lower (66%), particularly for mobile networks, NATed users, and VPNs. We do not assume either system is perfect. Instead, we treat agreement between them as a sign of correctness: if both independently place the client in the same metro, confidence in that location increases. To check this, 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 (e.g., VPN use, mobile NAT, or misannotated prefixes).

Impact of filtering. 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. After filtering, we retain a dataset where user groups reliably reflect metro-level client locations, reducing noise caused by incorrect IP-to-location mappings. By focusing exclusively on measurements with consistent geolocation between the two systems, we ensure that metro-level analysis and attribution in this paper are based on a high-confidence subset of clients.

B.2 Adding metadata to traceroutes

IP to AS: We use the same method as prior work [93] to map the IP addresses from traceroutes and reverse traceroutes to their AS: We prioritize IXP data from PeeringDB, PCH, and Hurricane Electric [1, 40, 71] over RouteViews [80]. Tools like bdrmapIT [59] improve IP-to-AS mapping by leveraging external dataset (e.g., alias resolution datasets), but at the $\langle \text{AS}, \text{metro} \rangle$ granularity used in this paper, these improvements rarely affect end results. Using CAIDA's ITDK dataset [11], we find that over 99.6% of interfaces map to the same $\langle \text{AS}, \text{metro} \rangle$ pair using both our approach and bdrmapIT, and fewer than 0.02% of paths differ by more than one hop. Because HERMES performs detection and attribution at AS and metro granularity, integrating bdrmapIT would add substantial complexity without improving localization accuracy.

IP to Facility: We map each hop to a possible facility using PoP-level data retrieved from iGDB [2].

IP to geolocation: We use the same method as prior work [83] to map each hop to its geolocation (latitude, longitude, city). We prioritize Hoiho [55] over RIPE IPMap [79], 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 [27]. To improve reliability, we finally 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, a well-known low-orbit Internet provider 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.

B.3 Detecting throughput anomaly

This section describes in greater detail our approach to detecting and analyzing anomalous throughput patterns using the Wasserstein distance, the Mann-Whitney-U test, the median difference, and the anomaly ratio. At a high-level, the methodology integrates weekly throughput data from client-server measurements to define a baseline. Our methodology uses a suite of metrics, each designed to highlight a distinct facet of throughput deviations. Below, we outline the purpose and justification for each metric, emphasizing their complementary roles in understanding throughput anomalies:

Median difference

The median difference quantifies how daily median throughput deviates from the weekly median, capturing shifts in the typical user experience. A larger difference indicates a more significant change in performance.

Wasserstein distance

The Wasserstein distance, also known as Earth Mover's Distance, measures the "effort" required to transform one distribution into another. The metric provides a holistic view of changes in throughput distribution, accounting for shifts in both shape and spread, and is particularly valuable for identifying distributional anomalies, such as the emergence of a heavier tail indicative of increased low-throughput users,

complementing the centroid and median-focused metrics. Unlike metrics such as Kullback-Leibler divergence, which are sensitive to small probability values and require overlapping supports, Wasserstein distance is robust to distributions with disjoint supports and provides interpretable results as a single unified measure of difference.

More formally, given two discrete probability distributions $P = \{(x_i, p_i)\}$ and $Q = \{(y_j, q_j)\}$, where x_i and y_j are discrete points (*i.e.,* bins of throughput values) and p_i and q_j are the corresponding probabilities (weights) at each point, satisfying:

$$\sum_{i=1}^{n} p_i = \sum_{j=1}^{m} q_j = 1,$$

let d_{ij} be the absolute difference $|x_i - y_j|$, we define the L^1 Wasserstein distance (also called Earth Mover's Distance) as:

$$W(P,Q) = \min_{\gamma \in \Gamma(P,Q)} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} d_{ij},$$

where $\gamma_{ij} \geq 0$ is the "flow" (amount of probability mass moved) from point x_i to y_j and $\Gamma(P,Q)$ is the set of all valid flows satisfying:

$$\sum_{j=1}^{m} \gamma_{ij} = p_i \quad \forall i, \quad \sum_{i=1}^{n} \gamma_{ij} = q_j \quad \forall j.$$

We apply a permutation test to evaluate the significance of observed Wasserstein distance, as follows:

1. Compute the observed Wasserstein distance W_{obs} between the daily and baseline distributions.
2. Generate a "null distribution" by randomly shuffling the combined data from both distributions, then splitting it into two new groups, and computing their Wasserstein distances. We repeat this many times to see what distances we would get if there were no real difference between the two distributions.[7]
3. Compute the p-value as the fraction of null distances greater than or equal to W_{obs}.

When the Wasserstein distance is computed in the L^1 norm and for one-dimensional distributions, the problem becomes significantly simpler. Instead of solving a general optimization problem where mass must be moved while obeying flow constraints, the 1D Wasserstein distance can be computed using a greedy algorithm that leverages the natural order of the points. Specifically, if P and Q are sorted in ascending order of x_i and y_j, respectively, the optimal matching pairs the smallest remaining mass from P to the smallest remaining mass from Q.

[7]The null distribution is created by combining both the baseline and day-of-interest data to ensure that any observed difference is evaluated against a scenario where no real distinction exists between the two datasets. This approach avoids assuming that the baseline alone represents all possible variability and allows us to test whether the observed data deviates meaningfully from what could occur by chance in a combined dataset.

Lemma B.1. *Let $P = \{(x_1, p_1), \ldots, (x_n, p_n)\}$ and $Q = \{(y_1, q_1), \ldots, (y_m, q_m)\}$ be two discrete probability distributions on $\mathbb{R}$, where p_i and q_j are probability masses such that $\sum_i p_i = \sum_j q_j = 1$. The Wasserstein-1 distance (Earth Mover's Distance) in the L^1-norm is defined as:*

$$W_1(P,Q) = \inf_{\gamma \in \Pi(P,Q)} \sum_{i,j} \gamma_{ij} |x_i - y_j|$$

where γ_{ij} represents the mass transported from x_i to y_j, and $\Pi(P,Q)$ is the set of all valid transport plans satisfying:

$$\sum_j \gamma_{ij} = p_i, \quad \sum_i \gamma_{ij} = q_j, \quad \gamma_{ij} \geq 0.$$

Then an order-preserving (greedy) matching is optimal.

Proof. Without loss of generality, assume that both P and Q are sorted in increasing order of their locations: $x_1 \leq x_2 \leq \cdots \leq x_n, \quad y_1 \leq y_2 \leq \cdots \leq y_m$.

In the 1D case, the optimal transport plan has a monotonic structure, meaning we should transport mass in an order-preserving manner: smaller values in P to smaller values in Q, and so on.

We construct an incremental transport plan by iteratively pairing the smallest available mass in P to the smallest available mass in Q. We define cumulative distribution functions:

$$F_P(x) = \sum_{i: x_i \leq x} p_i, \quad F_Q(x) = \sum_{j: y_j \leq x} q_j.$$

The optimal transport plan follows from the observation that in 1D, the transport cost is minimized when the cumulative masses of P and Q evolve together. The mass movement satisfies:

$$W_1(P,Q) = \int_{-\infty}^{\infty} |F_P(x) - F_Q(x)| dx.$$

We define the following greedy procedure:

- Start with $i = 1, j = 1$.
- Transport mass $\min(p_i, q_j)$ from x_i to y_j.
- Update the remaining masses: $p_i := p_i - \min(p_i, q_j), \quad q_j := q_j - \min(p_i, q_j)$
- If $p_i = 0$, increment i; if $q_j = 0$, increment j.

It is easy to see that the total transport cost of this algorithm is: $\sum_{i=1}^{n} |F_P(x_i) - F_Q(x_i)|$ which is exactly equal to $W_1(P,Q)$. □

Anomaly ratio The anomaly ratio measures the fraction of daily throughput values falling below our base (*i.e.,* median of the weekly throughput). This metric directly quantifies

the user impact of low-throughput events, offering a practical perspective on the scale of performance degradation.

Mann-Whitney U Test The Mann-Whitney U test is a non-parametric statistical test used to compare two independent distributions, determining whether one tends to have higher or lower values than the other. Unlike parametric tests, it does not assume a normal distribution, making it robust to non-Gaussian data.

Null Hypothesis (H_0): The daily and baseline throughput distributions are drawn from the same population (i.e., no shift in central tendency).

Alternative Hypothesis (H_1): The daily throughput distribution has shifted compared to the baseline (i.e., there is a significant difference in rank distributions).

Test Statistic (U): The test ranks all observations from both distributions together and computes the sum of ranks for each group. The statistic U is derived from these ranks and compared against a distribution under H_0 to compute the p-value.

Bringing it all together: In Figure 9, we present a comprehensive visualization of our multi-dimensional anomaly detection framework. The upper panel juxtaposes the throughput distributions of the baseline (in blue) and daily data (in orange), using shared binning where bin edges are determined from the combined range of both datasets to ensure direct comparability between the two datasets. Overlaid on the baseline histogram is a fitted GMM, emphasizing the dominant modes within the weekly distribution. To further aid interpretation, we include a vertical line marking the median of the baseline throughput, as well as annotations detailing the number of measurements for both the baseline and the anomaly day, along with the Wasserstein distance between the two distributions. Notable deviations between the orange and blue histograms, particularly reductions in density at higher throughput values or increases at lower values, signal potential performance degradations. The Wasserstein distance serves as a succinct metric capturing the overall magnitude of these shifts, providing a quantitative summary of the observed divergence.

The middle panel focuses on the metrics we use to detect anomalies, comparing their observed values against established thresholds. These include the Mann-Whitney and Wasserstein p-values, the percentage median decrease, and the fraction of points above thresholds. Bars are color-coded to enhance interpretability: green indicates the metric does not meet the anomaly threshold, while red highlights metrics that flag anomalous behavior.

Finally, the lower panel explores the alignment of daily throughput values with the baseline GMM centroids, providing a detailed view of how the distributions differ. The heatmap's first row represents the observed probabilities of daily throughput values aligning with each baseline GMM centroid, as well as their likelihood of being classified as "No Fit." The second row shows the baseline GMM component weights for reference. Discrepancies between these rows, such as elevated probabilities in the "No Fit" column or significant shifts in alignment across centroids, indicate structural changes in the throughput distribution. This panel is included for illustration only and is not part of the process described in Section 4.1.3.

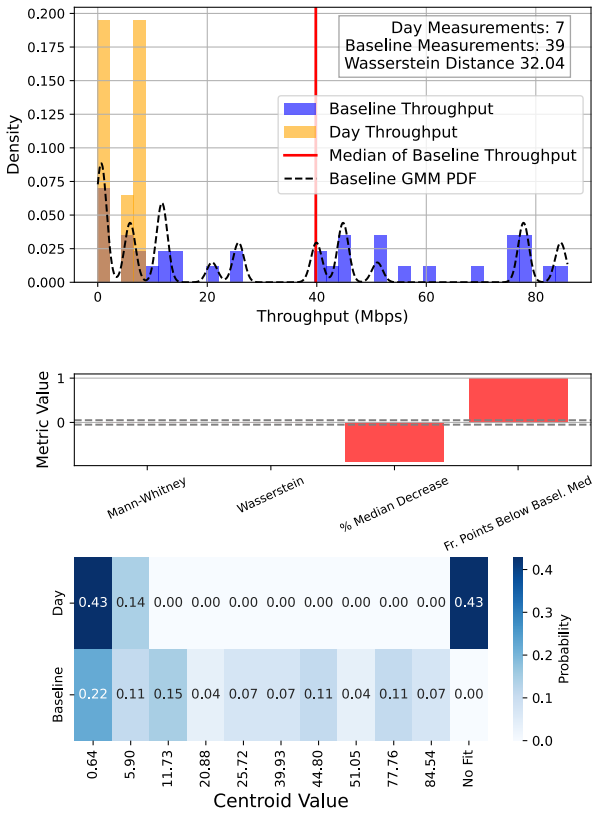


Figure 9: Visualization of throughput anomalies between ⟨AS8151, Coquimatlan-MX⟩ and a server in Mexico. The red vertical line marks the baseline median throughput, and annotations highlight the Wasserstein distance (32.04) and the number of measurements (7 for the day, 39 for the baseline). The daily distribution shows increased density at lower throughput values, indicating a degradation in performance. The middle panel evaluates anomaly detection metrics: Mann-Whitney and Wasserstein p-values are below the threshold, while the percent median decrease and fraction of points above the threshold indicate clear signs of an anomaly ongoing. The bottom panel is included for illustration purposes, showing how daily measurements align (or fail to align) with baseline GMM centroids to highlight structural shifts in the distribution.

B.4 Correlation Tomography details

To make the *Correlation Tomography* more explicit, Algorithm 1 outlines the steps of our correlation tomography procedure. The algorithm iteratively filters unexplained anomalies, identifies likely culprits by comparing anomalous and non-anomalous paths, and updates the set of explained events. It then aggregates evidence across multiple dimensions (AS,

metro, IXP, facility) and assigns the most plausible explanation for each event.

B.5 Details for probability matrix and additional measurements

A central component of metAScritic [83] is a probability matrix P that estimates, for each vantage point v and target IP t, how likely a traceroute will traverse a candidate link ℓ. In this section, we describe how we repurpose $\mathbb{P}$ to guide probe selection.

Repurposing $\mathbb{P}$ for probe planning. Suppose an ambiguity set $S = \ell_1, \ell_2, \ldots, \ell_k$ arises from tomography, meaning that HERMES is not able to identify which of those entities is responsible. Our goal is to choose traceroutes that maximize expected information gain about which $\ell_i \in S$ is responsible for the anomaly. For each candidate measurement from site s to user group **UG**, we use the approach from metAScritic to compute the probability vector

$$\mathbf{p}_{s,\mathbf{UG}} = \big(\mathbb{P}(s,\mathbf{UG},\ell_1),\ldots,\mathbb{P}(s,\mathbf{UG},\ell_k)\big)$$

Where $\mathbb{P}(s,\mathbf{UG},\ell_1)$ denotes the probability that a traceroute from the vantage point s to a destination in **UG** cross ℓ_1 as given by metAScritic's estimator. The informativeness of $(s,\mathbf{UG})$ is scored by a separation function $I(\mathbf{p}_{s,\mathbf{UG}})$:

$$I(\mathbf{p}_{s,\mathbf{UG}}) = 1 - H(\mathbf{p}_{s,\mathbf{UG}})$$

where $H(\cdot)$ is the normalized entropy. Intuitively, a site-user group pair is more informative when its probability vector strongly favors one link over the others resulting in an entropy close to 0.

Greedy selection algorithm. We iteratively select the $(s,\mathbf{UG})$ pair that maximizes expected reduction in ambiguity scaled by the impact of the ambiguity set:

$$(s^*,\mathbf{UG}^*) = \arg\max_{(s,\mathbf{UG})} w(\mathbf{UG})\Delta(S \mid (s,\mathbf{UG}))$$

where $\Delta(S \mid (s,\mathbf{UG}))$ denotes the expected decrease in the size of S after probing $(s,\mathbf{UG})$, and $w(\mathbf{UG})$ reflects the number of users in **UG** and the severity of degradation associated with **UG**. The process repeatedly selects $(s^*,\mathbf{UG}^*)$ until the probing budget B is exhausted. After each measurement, $\mathbb{P}$ is updated with posterior probabilities as described in the original work [83].

B.6 Why HERMES does not use loss

We exclude packet loss as a detection signal in HERMES because it is noisy and difficult to attribute at Internet scale. Observed loss is often dominated by transient effects in home Wi-Fi or mobile networks, making it a poor indicator of ISP- or backbone-level problems. Loss reporting in tools like NDT is coarse, heavily influenced by congestion control dynamics, and exhibits bursty, heavy-tailed distributions that complicate aggregation. Moreover, severe loss typically manifests as throughput degradation, which HERMES already measures. Given these limitations and the high operational cost of validating loss anomalies, we defer its integration to future work.

B.7 Variance in throughput versus latency

We analyze the variance difference between throughput and latency using all speed tests collected since January 1, 2025. To ensure comparability, we apply min-max normalization to latency and throughput for each client IP at every M-Lab server. To reduce noise, we exclude clients with fewer than 10 measurements per site before computing the variance of each metric. Clients are then categorized based on whether latency exhibits less variability than throughput. Among the 6.1 million observed IP-to-M-Lab pairs, throughput showed higher variance in most cases, while 2.2 million pairs had greater latency variance. When aggregated at the $\langle \text{metro}, \text{AS} \rangle$ level, the trend is even more pronounced, with 76.8% of user groups experiencing higher variance in throughput than in latency.

C Evaluation Details

C.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. We put an example in Figure 10.

Mailing lists: The NANOG and the Outages mailing lists [65, 67] 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 $\langle \text{ISP}, \text{metro} \rangle$ 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 [76] to crawl the following subreddits from the 1st of July 2024 onwards:

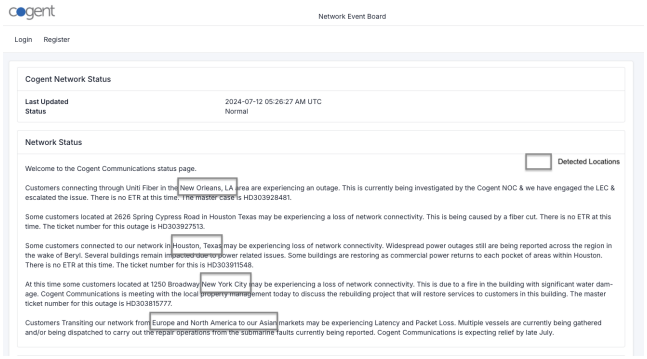


Figure 10: Archived Cogent Network Status Page. This page provides real-time (at the time of the archive) updates on network incidents and outages impacting customers in various regions. The status report includes details such as affected locations (*e.g.,* New Orleans, Houston, New York City, and international connectivity between Europe, North America, and Asia), outage causes (*e.g.,* fiber cuts, power outages, fires, and submarine cable faults), and associated ticket numbers for tracking.

```
['ATT', 'Comcast', 'Verizon', 'CoxCommunications', '
    Spectrum', 'centurylink', 'FrontierFios',
 'optimum', 'Suddenlink', 'Windstream', 'GoogleFiber', '
    RCN', 'Mediacom', 'MetroPCS',
 'TMobile', 'Sprint', 'CricketWireless', 'bell', 'Rogers
    ', 'Telus', 'ShawCommunications',
 'Videotron', 'SaskTel', 'Eastlink', 'BTCommunity', '
    VirginMedia', 'SkyBroadband', 'TalkTalk',
 'Plusnet', 'EE', 'Telstra', 'Optus', 'TPG\_Telecom', '
    iiNet', 'AussieBroadband',
 'Vodafone_Australia', 'SparkNZ', 'telekom', 'KPN', '
    Ziggo', 'FreeMobile', 'Singtel', 'NTT',
 'SoftBank', 'Vodafone', 'Orange', 'TMobile', 'Telefonica
    ', 'techsupport', 'ISPcomplaints',
 'outages', 'sysadmin', 'HomeNetworking', 'networking']
```

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, such as:

```
[
    'internet outage', 'network down', 'no internet', '
        slow internet', 'latency issues',
    'packet loss', 'connection problems', 'wifi issues',
        'modem reset', 'disconnecting',
    'service interruption', 'slow internet speed', 'speed
        test', 'internet interruptions',
    'internet problems', 'internet service', 'internet
        connection', 'internet speed'
]
```

Additionally, for regional subreddits, we expand the search to include 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 [70] with a structured prompt (see Appendix C.3). It extracts details such as the nature of the issue, affected networks, and ISPs. The results are filtered and saved as a refined dataset for analyzing user-reported network problems. For the vast majority of the instances, the ASN is not explicitly mentioned, so we extract the relevant ASNs using PeeringDB [1]. By the end of the process, we obtain a total of 3,207 posts where we detect events. Figure 11 shows the distribution of the subreddits these posts originate from, while Figure 12 highlights the networks associated with these events.

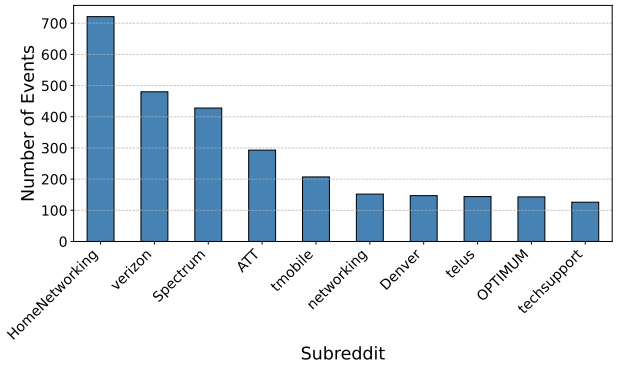


Figure 11: Distribution of the most frequently mentioned subreddits discussing detectable Internet events.

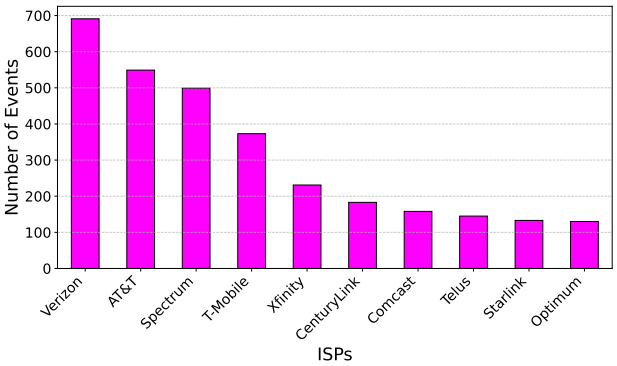


Figure 12: Distribution of the most frequently discussed networks for detectable Internet events.

C.2 Recall evaluation

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: From the collected dataset, we manually reviewed 288 Reddit posts and found that 213 (74.0%) described actual network events. Within this subset, 138 (64.7%) were observed by HERMES. 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. Across the full dataset, HERMES detected 57.2% of the presumed valid events. 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—12× less than what we had.

C.3 LLM prompts

The extraction prompts were refined through trial and error to maximize classification accuracy, and improving them further through systematic prompt engineering is an avenue for future work.

```
Given the following thread, please extract the relevant
     details:


Message: {message}


Please extract:
- Date (e.g., "Date: 2024-03-10")
- ASN (e.g., "ASN: 12345")
- Network Impacted (e.g., Meta, Google)
- Protocol (e.g., "Protocol: BGP, DNS")
- Point of Presence (e.g., "PoP: XYZ, City, Country")
- Lat-Lon of PoPs (e.g., "PoP Lat-Lon: 12.34, 56.78")
- Location Impacted (e.g., "City, Country")
- Lat-Lon of Impacts (e.g., "Lat-Lon: 12.34, 56.78")
- Subject or Problem description
  (e.g., "Subject: Route missing")
- Network Problem Description
  (e.g., "Problem: route not visible")
- Is this a problem? (Yes/No)
- Hour (e.g., 10:00)
- Raw message content in full string format
  (i.e., the full email content)


If there is a list of networks, make sure to put it
in a list format. Please have the Country in ISO2 format.
If there is no lat-lon, infer it from the City, Country.
Please keep the format identical! Return this as JSON
     format.



You are an assistant extracting structured data about
potential internet/connectivity issues from a Reddit
discussion thread.


Thread: {thread_text}


Please provide a JSON object with the following fields:
{
 "Is Internet Issue?": "Yes/No",
 "Is Visible with Speed Test?": "Yes/No",
 "Problem Description": "string describing the issue",
 "Location": "City, Country if available",
 "ASN": "List of Autonomous System Numbers",
 "Network": "List of Networks Mentioned",
 "ISP": "List of ISPs Mentioned",
 "Protocol": "List of Protocols Mentioned",
 "Points of Presence Affected": "List of PoPs Affected",
 "Thread Summary": "summary of the conversation",
 "Raw Thread": "the entire thread text"
}

If there's no obvious internet issue, set
"Is Internet Issue?" to "No".
```

C.4 More aspects of coverage

In Section 5.3, we centered our discussion on logical coverage, emphasizing HERMES's ability to observe a sufficiently comprehensive view of the topology to detect events that materially impact end-user experience. However, coverage is inherently multidimensional and must also encompass (i) population-centric metrics and (ii) infrastructure-level visibility, extending beyond the ISP "eyeballs" perspective introduced earlier. This section examines these two dimensions in greater depth.

When we discuss coverage in the context of HERMES, our primary goal is to ensure that HERMES can detect events that significantly impact end-users, as its design prioritizes user experience above all else. If an event at a PoP, for example, does not propagate downstream in a way that alters user performance, we do not expect to observe it. We also recognize the importance of infrastructure-level coverage. For instance, even if a specific user group is not directly measured, full visibility into its upstreams can provide indirect coverage by detecting outages or disruptions that affect the paths that cross them. To quantify this, we analyze HERMES's ability to cover PoPs and IXPs using PeeringDB [1] and AS Rank [12].

Additionally, geographic coverage plays a critical role in ensuring HERMES adequately represents global user experiences. Internet usage is not uniform across regions, and network performance issues often have localized impacts. A system that disproportionately measures certain areas while neglecting others risks providing a skewed view of the Internet's state. To address this, we evaluate geographic coverage with a focus on ensuring alignment between population distribution and test density (Appendix C.4.1). By examining the relationship between metro area populations and the frequency of measurements, we aim to assess whether HERMES captures a representative view of end-user experiences across different regions.

C.4.1 Geographic Coverage of HERMES

We compare HERMES's speed test counts across a week to the population size of metro areas, using satellite-derived population estimates [96] combined with iGDB's Voronoi's cell [2] to delineate metro boundaries and aggregate their population totals. To visualize this relationship, we generate an anamorphic map (Figure 13). 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

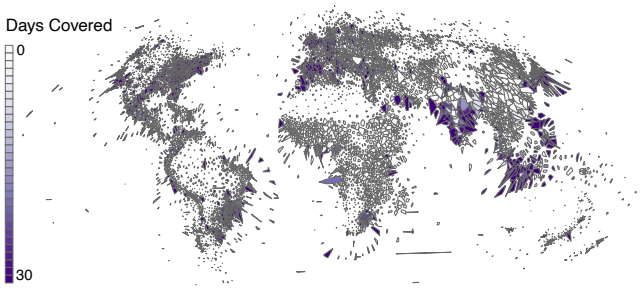


Figure 13: Anamorphic map showing the number of days with sufficient network coverage across metro areas. Each cell is scaled by population, with color indicating the average days covered per month.

have the most significant impact on 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 confirms that many large cells in the map are shaded in purple, but there exist still regions with few speed tests compared to the population, such as Russia, China, some zones in India and some zones in Africa.

C.4.2 Infrastructure Coverage of HERMES

Infrastructure coverage—monitoring upstream paths and facilities—is an important aspect of topology monitoring. Even if HERMES lacks direct measurements from a specific user group, it can still achieve indirect coverage by monitoring all upstream paths that serve this user group. In such cases, HERMES can infer outages and performance degradations affecting end-users without direct measurements. To quantify this middle-path coverage, we evaluate the fraction of metros, PoPs, ASes, and IXPs covered by our measurements, using iGDB [2] as a reference for the geographic components and AS Rank for the logical ones [12]. Additionally, we compare our coverage with ITDK, which provides a more complete view of interconnection infrastructure through extensive active probing. While ITDK's numbers will be more comprehensive due to its methodology, this comparison offers valuable context for understanding the reach of HERMES's observations.

Table 4 shows that HERMES observes over 3,100 ASes, nearly 7,600 metros, and 201 IXPs globally. While this is fewer entities than those covered by Internet-wide efforts like ITDK, HERMES achieves broad geographic reach: it observes traffic from 239 countries (96% of all countries), demonstrating that end-user measurements provide visibility into nearly every region of the Internet. Overall, this comparison reinforces that HERMES is not intended as a full Internet topology census—tools like ITDK serve that purpose—but as a performance-focused monitoring platform. By combining broad geographic reach, representative coverage of major access and transit networks, and direct measurements from real user traffic, 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	201 (16.7%)	614 (50.9%)	1205
Count of $\langle ASN, IXP \rangle$	1,379 (2.0%)	18,044 (26.6%)	67,797
Count of ASN	3,119 (4.0%)	67,620 (87.1%)	77,642
Count of Metro	7,587 (9.1%)	30,520 (36.7%)	83,218
Count of Country	239 (96.0%)	248 (99.6%)	249
Count of $\langle ASN, Metro \rangle$	17,811 (31.4%)	42,546 (75.1%)	56,663

Table 4: Comparison of Internet entities viewed by HERMES, ITDK, and the whole Internet [11]. Percentages are relative to the whole Internet.

C.5 Sensitivity analysis

We evaluate the robustness of HERMES's detection pipeline by systematically varying its key hyperparameters and quantifying their effect on event detection. Our goal is to demonstrate that the methodology is not overly sensitive to arbitrary parameter choices. In particular, we pick 3 successive days in December 2024 and we examine:
Significance thresholds: We sweep the p-value used for Welch's t, Mann-Whitney U, and 1-Wasserstein permutation tests from 0.01 to 0.10 (10 values). **Sensitivity thresholds:** We vary latency sensitivity ε_{RTT} from 1-20,ms and throughput sensitivity δ_{TP} as a *relative* drop from 5-40 (optionally, an alternate pass uses an *absolute* drop of 2-10,Mbps). **Majority rule:** We change the required fraction of measurements that must cross the threshold from 60-95 (10 values). **Baseline length:** We evaluate rolling windows of $1, 3, 5, 7, 10, 14, 21, 30, 60$ days. **User-group coverage:** We vary minimum coverage from $1, 3, 5, 7, 10, 20, 25, 50, 100$ distinct IPs and $3, 5, 10, 15, 25, 50, 100, 200, 500$ tests per (ASN, city, site) per week. **Dominance cap:** We limit any single IP's contribution to $10\%, 20\%, 30\%, 50\%$ of a group's measurements.

Evaluation metrics. For each configuration, we measure the total events detected according to each (latency and throughput) and match fraction between detections under each sweep and the baseline configuration.

Findings. Across all sweeps, HERMES's detection is stable under reasonable hyperparameter choices. The p-value sweep

shows smooth changes around the default of 0.05 (Figure 14). Decreasing the throughput threshold or increasing the latency threshold lowers the number of detected events, as expected. Likewise, raising the majority requirement makes the criteria stricter, since a larger fraction of measurements must exceed the threshold for an event to be flagged (Figure 15). Applying an IP dominance cap prevents any single client from disproportionately influencing results, and raising the minimum number of tests IPs reduces coverage as expected, reflecting the trade-off between detection ability and accuracy (Figure 16). Surprisingly, varying the baseline window length has a strong effect on the number of detected events (Figure 17). This large swing arises because changing the window also changes the set of user groups that meet coverage, so more (or fewer) groups contribute to the counts. When we restrict to the same subset of user groups and compare their baseline medians directly using CDFs, the differences across windows are much smaller.

Taken together, these results demonstrate that HERMES is resilient to reasonable parameter choices and that our defaults are relatively balanced. More importantly, these parameters are one of the many that could be selected and hyperparameter tuning could result in improved performance. We leave this effort as future work.

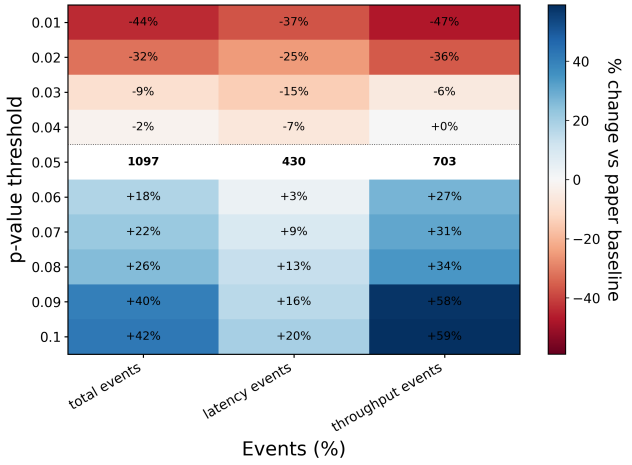


Figure 14: Effect of statistical significance threshold on event counts. Each cell shows the percent change in detected events (total, latency, throughput) relative to the paper baseline (p=0.05, dotted separator). Looser thresholds (bottom) increase detections gradually, while stricter thresholds (top) reduce them, indicating stability around the default.

C.6 Adding more measurement to help pinpointing

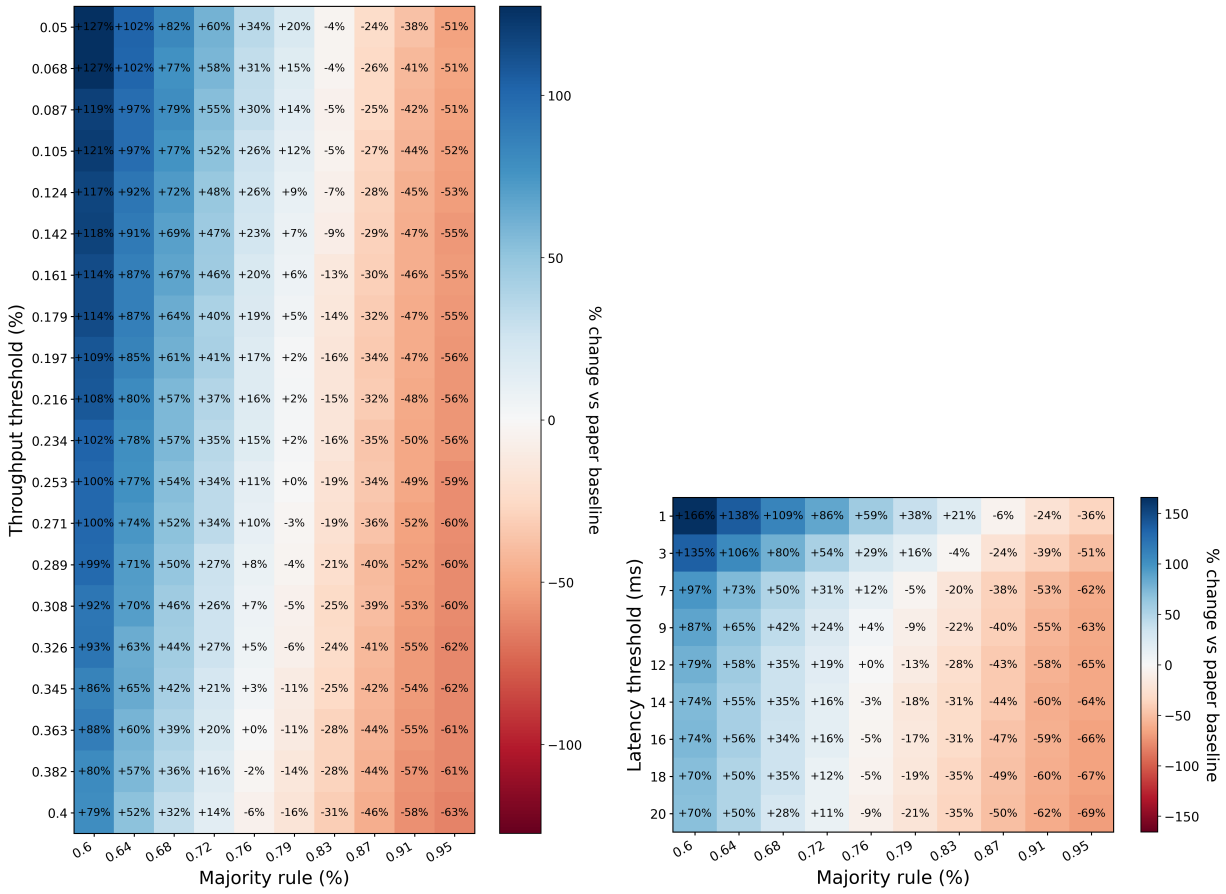
When HERMES detects an event, using only the topology measurements available often leave ambiguity sets of edges that could all plausibly explain the observed performance degradation. To reduce this uncertainty, we augment the dataset with additional target measurements as discussed in Section 4.4. To evaluate the effectiveness of our probing mechanism, we measure the reduction in ambiguity set size before and after incorporating targeted measurements. We allocate a budget of 10K reverse traceroutes per batch, spaced every two hours, totaling 120K measurements daily. We select each batch according to the algorithm described in Section 4.4. These targeted traceroutes reduce the median ambiguity set size by 47%, from 2.4 (edges) to 1.7 (edges). In 31% of cases, the additional measurements fully resolve ambiguity, isolating a single edge as the likely source. The impact is even greater for user groups with frequent anomalies, where prior data further constrains possible explanations, leading to 52% of them being fully identifiable.

We also compare the approach to a baseline where measurements are randomly selected without considering their potential to resolve ambiguity. This random selection reduces ambiguity in 75% fewer cases than our targeted approach, demonstrating the importance of strategically choosing measurements.

While adding traceroutes significantly improves our ability to identify the source of network events, it does not fully resolve all cases, and we identified more than 15K events where no measurement would have reduced the ambiguity. These findings emphasize the need for measurement infrastructure distributed across diverse networks to maximize HERMES's effectiveness.

C.7 Studying the number of speed tests per user groups during events

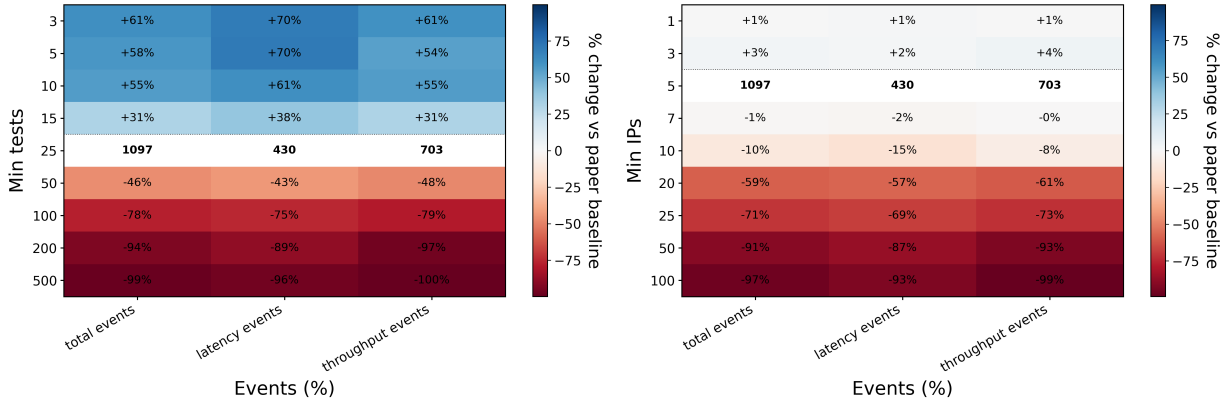
To test whether users are more likely to run speed tests when experiencing degraded performance, we examine post-hoc whether user groups show increased measurement activity on days flagged as anomalous by HERMES. For each user group over a three-month period, we compare the number of tests on anomaly days to non-anomaly days using a one-sided Welch's t-test. We investigate whether users are more likely to run speed tests on days when their network performance is degraded compared to normal conditions. 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 observation corroborates our intuition that the measurement bias in our dataset points in the right direction: users are more likely to run tests when something is wrong. While we observe a correlation between anomalies and increased volume, we also find spikes in test volume that do not coincide with performance degradations, suggesting that volume alone is not a sufficient signal for detection and must be combined with other performance indicators to avoid false positives.



(a) Majority × Throughput threshold - Total events

(b) Majority × Latency threshold - Total events

Figure 15: Percent change in total events detected relative to the paper baseline (*Majority rule* = 0.80, *Latency threshold* = 5 ms, *Throughput threshold* = 20%). The first plot (a) sweeps **Throughput threshold (%)** on the *y*-axis vs. **Majority rule (%)** on the *x*-axis; The second plot (b) sweeps **Latency threshold (ms)** vs. **Majority rule (%)**. Each cell shows the % change (blue: more events; red: fewer)



(a) Effect of minimum test requirement

(b) Effect of minimum IP requirement

Figure 16: Sensitivity analysis of HERMES across robustness parameters. Increasing minimum tests or IP requirements sharply reduces coverage, showing the trade-off between robustness and inclusivity.

C.8 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 can be fully trusted, *i.e.,* those that reach their destination without relying

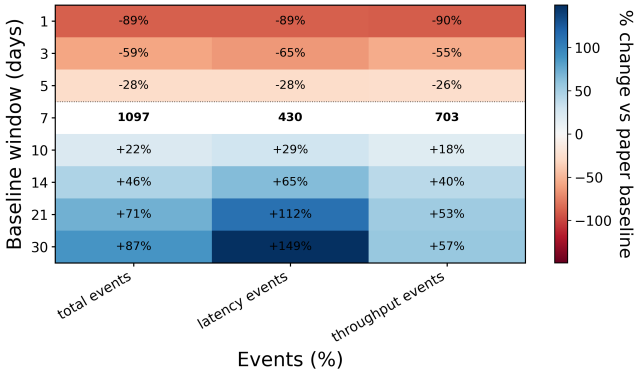


Figure 17: Sensitivity analysis of HERMES across robustness parameters. Varying the baseline window length impacts the number of events.

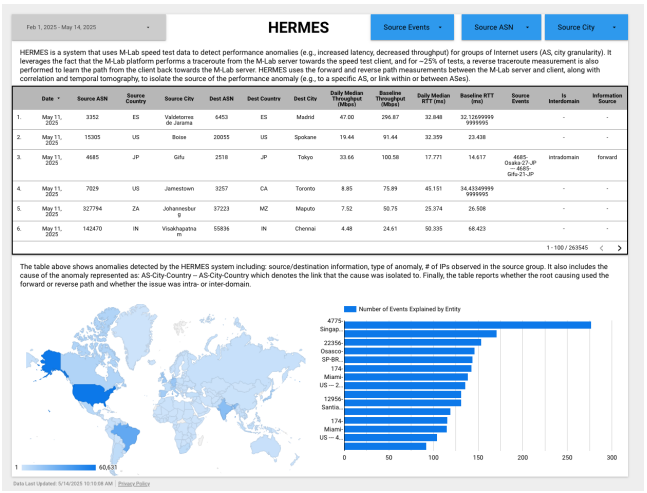


Figure 18: HERMES's aggregate interface showing detected performance anomalies across ASes and cities. The table at the top lists daily anomalies with information such as source and destination ASNs/cities, performance metrics (throughput, latency), and whether the issue was inter- or intra-domain. When available, the specific link attributed as the root cause is also shown. The map below visualizes the global distribution of events, while the bar chart ranks the most frequent root causes by AS or city.

on the interdomain asymmetry assumption [93].

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 do not rely on interdomain asymmetry, which accounts for approximately 35% of all reverse traceroutes, meaning they can be fully trusted. Compared to the most recent IMC study, where (i) was 58% and (ii) 31% of measurements could be fully trusted, our results indicate a similar overall measurement success rate but a slightly higher fraction of trustworthy paths.

D Complementary measurement studies with HERMES

D.1 Dashboards

Detecting performance anomalies at Internet scale is difficult, in part because robust ground truth is scarce and validating inferences across diverse vantage points is often infeasible. We acknowledge that some of the conclusions drawn by HERMES may remain unverifiable. However, since validation is not always possible, HERMES should provide transparency into why a particular anomaly is labeled and how it was attributed to a specific network component. To support this, we developed two complementary interfaces to make HERMES's inferences more interpretable. The first interface (Figure 18) offers an aggregate view of detected events, including high-level information like 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. The interface also highlights whether the anomaly was intra- or inter-domain and whether it appeared on the forward or reverse path. A bar chart on the right summarizes which ASes or metro regions explain the highest number of events over the selected time window.

The second interface (Figure 19) allows users to drill down into individual events. Operators and researchers can select a specific anomaly affecting a user group on a given day and inspect the associated traceroutes and speed tests that led us to the inference. The interface provides interactive visualizations, including: (i) time series plots showing the evolution of RTT and throughput relative to their baseline, (ii) logical path diagrams illustrating the sequence of ASes, IPs, and geographic locations traversed by packets, (iii) geographical maps showing the physical routing path of both forward and reverse traceroutes, and (iv) an interactive AS-metro topology graph. In this topology view, users can click on individual nodes or links to filter and inspect all measurements that traversed a given AS or interconnection in a specific metro. Users can also click on individual measurements within the plots to examine associated metadata and traceroute paths. Finally, the interface includes a feedback form that allows operators to flag misattributions or submit corrections, enabling continuous refinement of HERMES's event attributions.

D.2 Quantifying routing asymmetry

In Figure 20, 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

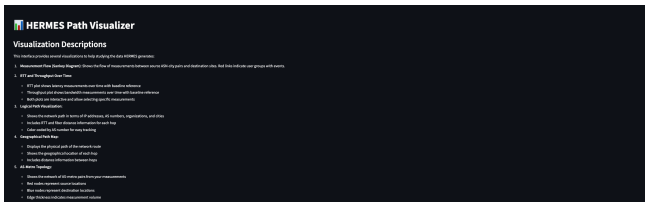


Figure 19: Event drill down and path visualization interface. This interface lets operator and researchers select a specific event (user group, server, day) and inspect the exact evidence behind the attribution. It provides (i) time-series plot for RTT and throughput, (ii) logical path diagram showing ASes, IPs, and metros crossed, (iii) geographic maps of forward and reverse paths, and (iv) interactive AS-metro topology that support node/link filtering to list all traversing measurements. Each plotted measurement is clickable to reveal metadata and traceroute paths. A built-in feedback form allows operators to flag misattributions or submit corrections, enabling continuous refinement of HERMES's attributions.

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, 99].

D.3 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 5 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 (e.g., Level 3 (Lumen), Tata Communications, Deutsche Telekom, and Vodafone Group). Prior work on persistent congestion relied on dedicated additional active measurements requiring to pre-select links or locations to probe [28], whereas HERMES leverages existing data to detect potential congestion points and supplements them with targeted traceroutes only when needed. This approach eliminates the need for specialized probe deployments and ensures that the focus remains on user-impacting congestion observed in the wild.

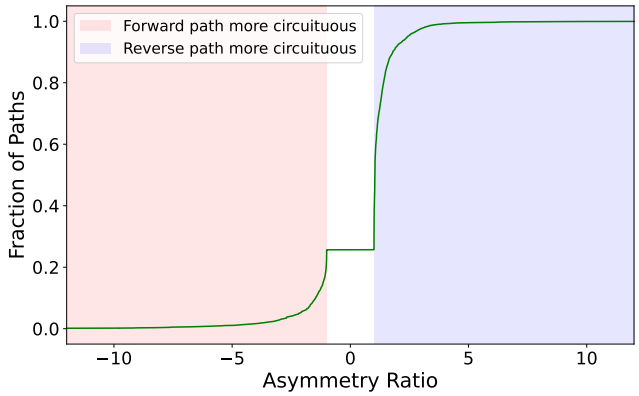


Figure 20: Distribution of asymmetry ratio for anomalous measurements. Negative values indicate the forward path is more circuitous, while positive values indicate the reverse path is more circuitous. For 72% of paths, the reverse path is more circuitous. For about 10% of the paths, the reverse path is at least twice as circuitous as the forward path.

Table 5: Persistently Congested Links.

Rank	Interconnection (ASN-City-Country)	Organizations
1	4755-Mumbai-IN - 9498-Mumbai-IN	Tata Communications - Bharti Airtel
2	1273-Milan-IT - 3356-Milan-IT	Vodafone Group - Level 3 (Lumen)
3	3320-Frankfurt am Main-DE - 3356-Frankfurt am Main-DE	Deutsche Telekom - Level 3 (Lumen)
4	3356-Rome-IT - 6453-Milan-IT	Level 3 (Lumen) - Tata Communications
6	1221-Brisbane-AU - 7575-Sydney-AU	Telstra - AAPT
7	3320-Frankfurt am Main-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 (Lumen) - Tata Communications
10	1267-Rho-IT - 6453-Milan-IT	Wind Tre S.p.A - Tata Communications

D.4 Weather and cable cuts

This section demonstrates HERMES's capability to analyze and visualize metro-level network anomalies caused by significant weather events and infrastructure disruptions. To evaluate network resiliency in these scenarios, we combine geographical data with anomaly detection metrics derived from HERMES. This enables the creation of event-specific maps that highlight how disruptions affect AS-level network behavior in impacted metros.

We focus on high-impact events, including severe flooding, hurricanes, typhoons, and undersea cable cuts, to assess the geographic distribution of network anomalies (Table 6 provides a detailed list of the events we considered). For each event, we query HERMES's anomaly detection database, using both pre- and post-event time windows to compute differences in the fraction of ASNs experiencing anomalies within affected metros. This approach isolates the impact of the event itself from normal variations in network behavior.

- **Valencia Flooding:** As shown in the upper left of Figure 7, the flooding produced a broad regional impact: many Valencian and nearby metros show large increases in the fraction of anomalous ASes (often > 0.4). The effect extends beyond the core flood zone, consistent with rerouting and shared infrastructure dependencies. This event represents the strongest, most spatially extensive signal among the cases we examined.
- **Typhoon Shanshan, Japan:** In the upper right Figure 7, the impact is comparatively limited. We observe very

Table 6: Summary of Events Considered

Date	Metro Impacted	Country	Description	Noticeable
2024-10-09	Florida Helene	US	Hurricane Helene made landfall, causing catastrophic flooding and widespread destruction.	No
2024-08-28	Tokyo-JP	JP	Typhoon-related flooding caused widespread damage to infrastructure and homes.	No
2024-10-29	Valencia-ES	ES	Massive flooding caused by torrential rains, leading to significant damage in the city and surrounding areas.	Yes
2024-10-17	Baltic Sea	EE, FI, LV, LT, PL, DE, SE, DK	Severe disruptions due to undersea cable cuts, impacting internet and communication infrastructure across the region.	Yes
2024-09	Andhra Pradesh	IN	Severe flooding caused widespread disruption, impacting more than 270,000 people.	Yes

modest, localized increases in anomaly fractions in a few metros along or near the storm's track, while most metros remain near baseline. This pattern suggests that network hardening and/or routing diversity constrained the typhoon's measurable performance impact.

- **Hurricane Milton, United States:** As illustrated in the bottom left Figure 7, we see elevated anomaly fractions tightly aligned with Milton's track through Florida and the Southeast. We also observe scattered increases in more distant U.S. metros, hinting at upstream dependencies or wider-scale traffic shifts. Together, these patterns suggest both direct storm impact and secondary effects from network-wide rerouting.
- **Baltic Sea Subsea Cable Cut:** The Baltic Sea, drawn on the bottom right of Figure 7, shows a very localized effect: a clear increase is concentrated at the Finnish landing point of the cable, with little to no change in other metros. The confinement of anomalies to the landing area indicates that the subsea cable here is unlikely to carry a lot of the traffic observed by HERMES.

E Ensuring Sustainability

With many large-scale measurement initiatives, a fundamental challenge lies not only in the design of the measurement mechanisms, but in ensuring the long-term sustainability of these efforts. The ecosystem of Internet measurement tools, platforms, and vantage points is shaped by an interplay between technical viability, financial support, operational maintainability, and critically stakeholder incentives. Historically, one of the great hurdles facing academic and open measurement platforms has been the absence of a natural incentive alignment that encourages all parties—users, operators, and researchers—to invest time and resources into keeping these projects alive.

If we investigate existing observatories (*e.g.,* PlanetSeer [97]), they often relied on goodwill from their participants and the hosting entities (*e.g.,* PlanetLab [20]). Without tangible returns—either monetarily via funding grants or in terms of academic accomplishments—these tools remain the domain of isolated research endeavors that prove challenging to maintain at scale. While CEM [19] stood out by directly improving user performance, it faced challenges in maintaining consistent participation over time as the popularity of P2P networks declined. Commercial entities, such as content providers or CDNs, may periodically share curated datasets, yet the metrics provided are inherently shaped by these organizations' internal business interests and whose methodologies are rarely open. By extension, this stranglehold results in a lack of transparency, unclear data representativeness, and insufficient incentive for these actors to extend their measurement infrastructures in ways that primarily would benefit the broader Internet community.

In contrast, M-Lab was founded to help address these issues and provide a platform for ongoing open Internet measurements. End-users run speed tests when they experience Internet issues, using them as diagnostic tools. The insights they gain—such as details about their connection quality—help them improve their service, hold ISPs accountable, or make better choices about their Internet providers. Because M-Lab's NDT test is embedded directly into Google Search—appearing as a built-in measurement tool within the world's most widely used search engine—it ensures consistent participation, keeping the dataset continuously fresh.

HERMES fills an important gap in the incentive loop: with our system, operators who are hosting M-Lab servers can detect large-scale events affecting their users and consequently, their business in a way that they could not without relying on complex custom-made detection schemes. This new view of the Internet enables them to proactively address performance issues, optimize their network, and improve user experience. By providing these benefits, HERMES encourages operators to support and contribute to the platform (by hosting more M-Lab sites for example), creating a fully closed incentive loop that benefits all the stakeholders.

Algorithm 1: Correlation Tomography

Input: Speed test data with forward and reverse paths, baseline performance metrics, anomaly thresholds α, β, γ, δ, κ sensitivty threshold ε

Output: Set of plausible culprits assigned to each event

Initialize total_anomalies to 0;
Initialize cumulative_anomalies_explained to 0;
Set iteration_count = 0;
Create an empty set plausible_culprits;
while *iteration_count* $<$ *max_iterations* **do**
    Increment iteration_count;
    Step 1: Filter unexplained anomalies;
    `Compute(` *unexplained_anomalies from data excluding previously explained src-dst pairs*`)`;
    if *no unexplained anomalies remain* **then**
        break;
    Step 2: Analyze paths for each $\langle$**AS**, **metro**$\rangle$;
    foreach $\langle AS, metro \rangle$ *link* **do**
        - $f^* \leftarrow$ Fraction of anomalous path passing per $\langle \text{AS}, \text{metro} \rangle$ link
        - $f \leftarrow$ Fraction of non-anomalous path passing per $\langle \text{AS}, \text{metro} \rangle$ link
        - $\rho \leftarrow$ Ratio of anomalous to non-anomalous path crossing link
    Step 3: Select links with highest anomaly ratio;
    `Identify(` *top* $\langle AS, metro \rangle$ *link with highest ratio of anomalous to non-anomalous paths with* $\rho > \varepsilon$`)`;
    Step 4: Update explained anomalies;
    `Update(` *cumulative_anomalies_explained with anomalies explained by the selected AS*`)`;
    Append the selected edges to plausible_culprits;
    if *cumulative_anomalies_explained covers all anomalies* **then**
        break;
Step 5: Aggregate culprits by $\langle$**AS**, **metro**$\rangle$ **node**;
`Aggregate(`*plausible_culprits by* $\langle AS, metro \rangle$ *, computing the fraction of links per AS that have anomalies*`)`;
Step 6: Apply thresholds on aggregated data;
foreach $\langle AS, metro \rangle$ **do**
    - If the overall fraction of anomalies across the metro exceeds β, mark the metro as suspect.
    - If the overall fraction of anomalous across the AS exceeds γ, mark the AS as suspect.
    - If the overall fraction of anomalies across the IXP exceeds κ, mark the IXP as suspect.
    - If the overall fraction of anomalies across the facility exceeds δ, mark the facility as suspect.
    - If the fraction of anomalous exceeds α, mark $\langle \text{AS}, \text{metro} \rangle$ as a suspect.
    - Else mark nothing
Step 7: Assign explanation to each event; **foreach** *incident* **do**
    Assign an explanation to the incident based on the classification from Step 6:
    - If neither end of the $\langle \text{AS}, \text{metro} \rangle$ edge is marked as a suspect, assign the explanation as `peering`.
    - If either end of the edge is marked, select the explanation with the highest priority, following this hierarchy:
        1. `metro`
        2. `IXP`
        3. `AS`
        4. `facility`
        5. `<AS, metro>`
return *Aggregated list of explanation to each event*;

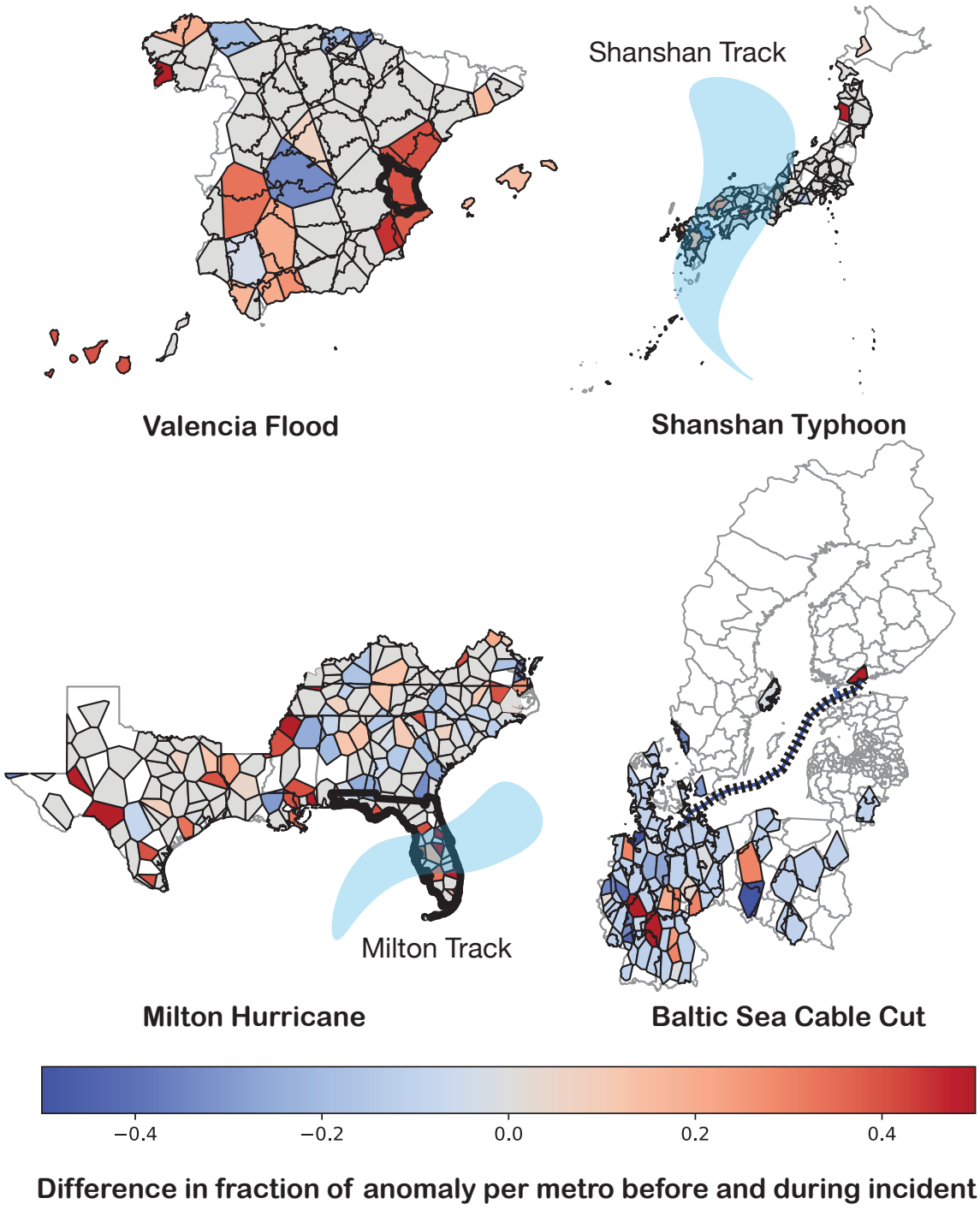


Figure 21: All events studied in Table 6.