USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

BayWatch: Practical Internet-Scale Topology Monitoring with Dynamic Bayesian Estimation

Zhongxu Guan, *Tsinghua University and Tsinghua Shenzhen International Graduate School;* Shuai Wang, Li Chen, and Zhaoteng Yan, *Zhongguancun Laboratory;* Jiaye Lin and Dan Li, *Tsinghua University;* Yong Jiang, *Tsinghua Shenzhen International Graduate School;* Yingxin Wang and Ziqian Liu, *China Telecom Cybersecurity Technology Co., Ltd.*

https://www.usenix.org/conference/nsdi26/presentation/guan-zhongxu

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

May 4–6, 2026 • Renton, WA, USA

ISBN 978-1-939133-54-0

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

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

BayWatch: Practical Internet-Scale Topology Monitoring with Dynamic Bayesian Estimation

Zhongxu Guan*‡, Shuai Wang†, Li Chen†, Zhaoteng Yan†, Jiaye Lin*,
Dan Li*, Yong Jiang‡, Yingxin Wang§, Ziqian Liu§
*Tsinghua University, †Zhongguancun Laboratory
‡Tsinghua Shenzhen International Graduate School §China Telecom Cybersecurity Technology Co.,Ltd.

Abstract

Internet topology monitoring is important for understanding topology dynamics. While a few commercial services have been provided to monitor the specified topology, academic research on Internet-scale topology monitoring still lags behind with two key limitations: 1) topology incompleteness caused by simplified assumption of uniform load-balancing responses (LBR) distribution; 2) low probing efficiency due to the lack of temporal awareness.

In this paper, we introduce BayWatch, a practical Internet-scale topology monitoring system that overcomes these limitations based on a Dynamic Bayesian Network (DBN). Leveraging the Markov property of packet forwarding, BayWatch models it as a sequence of state transitions over time within the DBN, so as to estimate the true LBR distribution and predict its temporal evolution. Internet-wide measurement results demonstrate that benefiting from the estimated LBR distribution, BayWatch can discover 2.4×/2.8× more nodes/links than the state-of-the-art algorithm, D-Miner, while the temporal awareness reduces the number of probes by 6.3× with negligible topology completeness loss. Moreover, we demonstrate that BayWatch can help detect anomalies using a real-world network outage event.

1 Introduction

The Internet has evolved into a critical global infrastructure at an unprecedented scale, supporting billions of users and mission-critical services. As its scale continues to grow, the demand for accurate and timely monitoring increases, especially topology monitoring, which is the foundation for detecting outages, diagnosing faults, and understanding global connectivity. Thus, many companies are commercializing monitoring products, such as SolarWinds Network Performance Monitor [35], LogicMonitor [29] , demonstrating both the technical feasibility and the business value of topology monitoring. However, these platforms largely target enterprise or datacenter environments, focusing on intra-domain topology and performance issues. At the Internet-wide level, only a few systems, such as Cisco's ThousandEyes [14], attempt to monitor dynamics of connectivity between ISPs and regions.

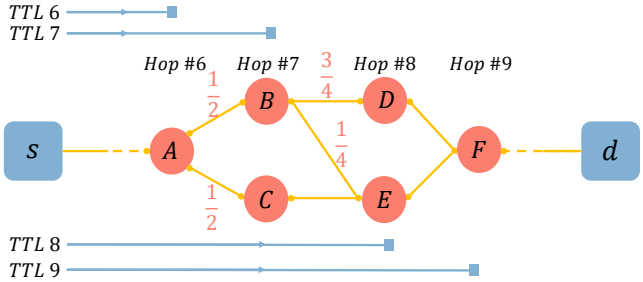


Figure 1: An example of multipath topology. With an uneven LBR distribution of $\{D{:}\frac{3}{4}, E{:}\frac{1}{4}\}$, the ratio of responses received by node B from nodes D and E is expected to be 3:1.

However, academic research still lags behind by focusing on topology *measurement* [1, 2, 11, 15, 19, 33, 40] rather than *monitoring*. Most of efforts address the challenge of revealing complicated multipath structures in the Internet, where load-balancing (LB) routers select the next hop for each data packet based on its header fields, as shown by router A in Figure 1. As a representative approach, Multipath Detection Algorithm (MDA) [8, 38, 39] sends multiple probes with random flow identifiers, which traverse different paths, to discover as many nodes and links as possible. The key assumption of MDA is that LB routers split traffic uniformly across next hops, which enables to compute the minimum number of probes required to statistically guarantee discovery of all successor routers and their corresponding links. However, existing topology measurement approaches face two fundamental limitations.

First, substantial portions of nodes and links remain hidden in the topologies they uncover, because the oversimplified uniformity assumption rarely holds in practice. For example, after probing 10^4 random prefixes in the Internet, we reveal that the assumption of a uniform LBR distribution holds in ∼20% of the cases. In particular, we utilize the well-known Gini coefficient (*GC*) [21] to quantify the uniformity of the LBR distributions obtained by our measurement, and find that 18% of the LB routers have a GC greater than 0.4, indicating significant unevenness. Using controlled experiments, we also

find that the state-of-the-art MDA, D-Miner [39], misses 24% of the nodes and 60% of edges[1] when the *GC* is 0.4. (§ 3.1)

Second, even when they manage to capture a reasonably complete topology in a snapshot, they still fall short of providing continuous visibility. The Internet is inherently dynamic. Important events, such as changes in routing policies, cannot be modeled by the existing MDA approach. As a result, independent snapshots struggle to distinguish genuine anomalies (*e.g.*, Internet outage) from ordinary variation. Moreover, because each snapshot attempts to rediscover the topology from scratch, as we show later, at least 86.7% of probes are wasted on revalidating edges that remain stable and known. (§ 3.2)

This gap highlights the need for approaches that can both capture more accurate multipath topologies and track their evolution over time. In this paper, we propose BayWatch, a practical Internet-scale topology monitoring system that uses a Dynamic Bayesian Network (DBN) to estimate the true LBR distribution and predict its temporal evolution. Based on our finding that packet forwarding usually follows a first-order Markov process, BayWatch models it as a sequence of state transitions over time within the DBN. By framing LBR distribution retrieval as estimating time-varying transition probabilities, BayWatch infers true LBR distributions from probe observations and predicts their evolution. With the transition probabilities estimated in previous snapshots, BayWatch achieves high completeness with reduced overhead by avoiding redundant validation of stable paths. Moreover, by comparing predicted distributions with actual measurements, BayWatch can distinguish topology changes caused by anomalies from routine fluctuations.

We first evaluate the ability of BayWatch to address the discovery incompleteness problem that arises from non-uniform LBR distributions. To this end, we select 100 topologies from the Topology Zoo [26] and set various LBR distributions for them as ground truth. The experimental results demonstrate that BayWatch uncovers more nodes and edges compared to D-Miner, identifying 85%∼90% of the nodes and 73%∼83% of the edges when $GC \leq 0.5$, which correspond to 89.4% of the LBR distributions characterized by our measurement. This improved performance is attributed to the ability of BayWatch to accurately estimate LBR distributions, particularly when *GC* is less than 0.5.

After validating BayWatch with the ground truth, we proceed to conduct Internet-scale measurements. We use 1 cloud virtual private server to run BayWatch, and collect 13 snapshots of the Internet topology, where each measurement requires 52 hours in the *bootstrap* mode and 7.8 hours in the *predictive* mode. In the first three snapshots, BayWatch operates without using historical data, discovering the topology from scratch. In this configuration, it discovers 2.5∼2.8 million nodes and 7.3∼7.9 million links per snapshot, which is about 2.4× and 2.8× the amounts discovered by D-Miner, respectively. D-Miner misses these nodes and links due to its assumption of a uniform LBR distribution, leading to premature detection termination. The subsequent snapshots are collected using the predictive mode, leveraging historical data as prior knowledge. We also run the bootstrap mode in parallel to provide a baseline for evaluating the predictive mode. The experimental results show that the predictive mode misses less than 1% of the topology discovered by the bootstrap mode.

With bootstrap mode, BayWatch requires 11.35 billion probes to obtain a complete topology snapshot, suggesting that achieving high-completeness topology discovery may require substantially more probes than previously estimated by existing methods. However, when operating in predictive mode, BayWatch becomes significantly more efficient. While maintaining higher topology completeness, it requires only about 40% of the probes used by existing methods, demonstrating its ability to avoid redundant probing. We further observe that BayWatch may miss a small portion of topology discovered by prior methods due to prediction inaccuracies; however, this portion is below 1%.

To illustrate how BayWatch can assist in anomaly detection, we present a case study of an event on the Zayo network [12]. BayWatch detected a sharp peak in the number of internal rehashing nodes on August 15, 2025. Our analysis confirms that this sudden change coincides with the period when the Zayo network experienced a service outage. This demonstrates that BayWatch not only discovers topological changes but can also provide internal signals that are useful for pinpointing real-world network failures.

Contributions. We make the following contributions:

- We investigate the uneven responses from routers' successors, and evaluate its impact on topology discovery. Our findings show that in about 20% of cases ($GC \geq 0.4$), D-Miner misses more than 60% of links.
- We propose BayWatch, an Internet-scale topology monitoring system centered around a Dynamic Bayesian Network, addressing the limitations of prior work: topological incompleteness and lack of temporal awareness.
- We validate BayWatch on controlled topologies and apply it to the Internet, discovering 2.4× and 2.8× more nodes and links than D-Miner. Even using only 40% of the probes required by D-Miner, BayWatch achieves 2.8× topology completeness, benefiting from the temporal awareness.
- We validate BayWatch with a large ISP, finding that 97.12% of the 108,752 links where the IP addresses of both endpoints belong to the ISP appear in their multipath topology. The remaining links are not found due to network dynamics.
- We show that BayWatch is a powerful tool for detecting real-world network anomalies.

[1] In this paper, we use the terms *link* and *edge* interchangeably.

- BayWatch is open-sourced at https://github.com/NASP-THU/BayWatch.
- The measurement results are updated monthly on the KI3 website [28].

This work does not raise any ethical issues, illustrated in detail in Appendix A.

2 Background

In this section, we provide background on multipath topologies and load balancing in the Internet. Then, we introduce Multipath Detection Algorithm (MDA), including the traditional MDA and the state-of-the-art method, D-Miner.

2.1 Multipath Topology of the Internet

In the Internet, a *path* is the sequence of IP addresses traversed by a data packet from its source s to its destination d. These IP address sequences uniquely identify each path, enabling the tracing of packet routes. To aggregate network capacity and provide resilience against failures, multiple paths often exist between the same source and destination [38].

Routers implement Equal-Cost Multi-Path (ECMP) routing, which distributes traffic across paths with equal cost metrics. These parallel paths typically reconverge before reaching the destination, forming a *diamond* structure between the divergence node (DN) and the convergence node (CN) [4]. Figure 1 illustrates an outer diamond, defined by the first DN and the final CN, which may itself contain nested diamonds.

Load balancing (LB) is crucial for utilizing multipath topologies. A common approach is per-flow LB, where all packets within the same flow follow the same path, while different flows are distributed separately. In this method, the router selects the next hop, *i.e.*, successor, for each packet based on a hash of header fields, typically including source and destination addresses, ports, and protocol, collectively referred to as the *flow identifier*. Other LB techniques are employed in the Internet. In per-packet LB, the router selects paths for each packet by rotating among successors in a round-robin fashion. In per-destination LB, packets destined for the same endpoint are directed to the same successor.

2.2 Multipath Detection Algorithm

Traceroute and its variants, such as Paris Traceroute [7], are commonly used to uncover the forwarding path from a source to a destination. These tools incrementally increase the time-to-live (TTL) values in ICMP probe packets and analyze the ICMP error responses from routers along the path. However, they primarily focus on improving the accuracy of identifying relationships between adjacent hops. In contrast, MDA is designed to discover all potential multipaths to a destination by varying flow identifiers [8, 38], offering a more comprehensive view of the network structure.

Traceroute-based MDA. MDA uses a round-based probing and calculates a statistically-guaranteed stopping point to determine the probes required in each round to discover all potential successors. For a DN with $k+1$ successors, MDA assumes that each successor has an equal probability of receiving a forwarded packet, leading to a uniform distribution of responses from successors, i.e., $1/(k+1)$. Based on this distribution, MDA calculates the stopping point, which is denoted as n_k. The stopping point is the number of probes with random flow identifiers required to ensure that the probability of missing one successor, after finding k of them, is less than $1-\alpha$, where α is the confidence level [38].

For instance, with TTL at 6 in Figure 1, assume that B is the only discovered successor of A after sending $n_1=6$ probes with random flow identifiers. The probability that C remains undetected is $(\frac{1}{2})^6=0.016<1-\alpha$. However, if both B and C are seen after sending $n_1=6$ probes, MDA will send $n_2-6=5$ more probes to discover any other successors. If no more successors are found, MDA concludes with 95% confidence that A has only two successors, B and C.

Yarrp-based MDA. Traditional traceroute-based MDA is slow because it probes each hop sequentially, waiting for a response before moving on. Vermeulen et al. [39] proposed D-Miner by combining MDA and Yarrp, which encodes state, including originating TTL and destination, into probe packets, thus allowing for stateless, parallel, and randomized probing. As a result, D-Miner can reduce the Internet-wide measurement from decades to 2 days.

However, D-Miner's concurrent, stateless probing requires pre-calculating the number of probes needed for each round to provide statistical guarantees. For example, fixing TTL at 7 in Figure 1 and assuming that 4 probes reached B and 2 reached C in previous rounds. Therefore, the probability distribution for the 7th hop $\mathcal{D}_7=\{\frac{4}{6},\frac{2}{6}\}$. D-Miner assumes this distribution will hold in the next round. To confirm that B has only one successor D, and C has only one successor E, $n_1=6$ probes should be sent to B and C, respectively. Based on the distribution of $\mathcal{D}_7$, a total of $\frac{6}{2}\times n_1=18$ probes should be sent at TTL 7 to ensure that at least n_1 probes pass through B and C, respectively. Then, these probes will be used by B and C to observe whether there are more successors in addition to the one already discovered.

In conclusion, while D-Miner employs stateless probing to accelerate measurements, it still relies on the MDA's uniform distribution assumption to provide its statistical guarantee.

3 Design Rationale of BayWatch

In this section, we overview the rationale for BayWatch's design to address the limitations of prior work. First, we show how we improve the coverage of multipath topology detection by challenging the simplistic assumptions that overlook nodes

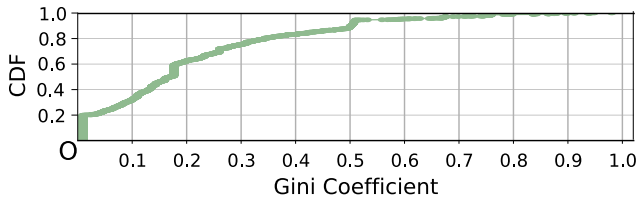


Figure 2: CDF of LBR distributions in the Internet.

and links. Then, we describe how BayWatch achieves efficient monitoring of the Internet topology.

3.1 Challenging the Uniformity Assumption

We first reveal that the assumption of a uniform distribution of responses from a DN's successors, *i.e.*, LBR distribution, frequently breaks down in today's Internet and explore the factors contributing to these uneven LBR distributions. Then, based on evaluations in controlled network environments, we demonstrate that these uneven LBR distributions significantly reduce the completeness of MDA's topology discovery.

3.1.1 LBR Distribution in the Internet

By sending a large number of probes with random flow identifiers, we can approximate the LBR distribution among a DN's successors. To analyze this in the wild Internet, we randomly select 10,000 IPv4 routable /24 prefixes from 268 distinct ASes as representative targets. For each prefix, we continuously send probes with random flow identifiers until the response frequencies from the different successor interfaces stabilize. Once stabilized, we record the frequencies as the LBR distribution.

We utilize the Gini coefficient (*GC*), a metric commonly used in Internet research to assess the uniformity of the distribution of various resources[21], to quantify the uniformity of these LBR distributions. *GC* is defined as follows:

$$GC = 1 - \frac{2\sum_{i=1}^{n}(n+1-i)x_i}{n\sum_{i=1}^{n}x_i} \quad (1)$$

where n denotes the number of a DN's successors and x_i denotes the i-th frequency in ascending order. The *GC* ranges from 0 to 1, where values closer to 0 indicate a more uniform distribution.

Figure 2 illustrates the CDF of LBR distributions in the Internet. We can see that only one fifth of the DNs receive responses evenly from their successors, satisfying the MDA's assumption, while 23% of the DNs have a GC of more than 0.3. Fortunately, only 10.6% of these DNs have a *GC* greater than 0.5, indicating that extreme distributions are rare. Appendix G further examines how the Gini coefficient relates to load-balancing behavior in Internet monitoring.

Causes of uneven LBR distribution. After investigating public deployment documents and engaging in private discussions with a prominent ISP, we conclude that three primary factors cause the uneven LBR distributions in the Internet. First, WCMP configuration [22, 45] adjusts traffic distribution

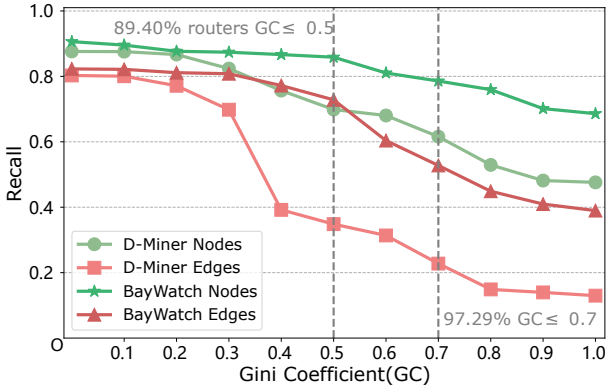


Figure 3: Recall on various LBR distributions.

based on link bandwidth differences to improve bandwidth utilization. Second, as a scalable solution to implement traffic engineering in large-scale backbone networks [23, 27], MPLS may result in uneven flow distribution across physical routers. Third, the unresponsiveness complicates the determination of the number of unresponsive nodes at each hop. Treating all non-responses as a special *dummy response* increases its probability, making it higher than that of any individual responsive successor. Examples caused by these factors can be found in Appendix B.

3.1.2 Impact of Uneven Distribution on MDA

We conduct an experiment to evaluate the impact of uneven LBR distributions on the performance of MDA in discovering topologies.

Experiment settings. We select 100 typical topologies from the Topology Zoo [26] as the ground truth. By adjusting the LBR distribution at each node, we vary the average *GC* of a topology in increments of 0.1, ranging from 0 to 1. This produces a set of topologies, each corresponding to 11 distinct *GC*s, with each *GC* represented by 100 distinct topologies. We then run D-Miner on these topologies and calculate the average recall for each *GC*. We define *recall* as the ratio of discovered nodes (or edges) to all nodes (or edges) in the topology. Note that we use D-Miner's open-source code and API [20] for the experiment.

Experiment results. As shown in Figure 3, D-Miner achieves high recall under uniform distributions ($GC \leq 0.1$), discovering 87% of nodes and 80% of edges. However, recall degrades quickly as *GC* increases. At $GC=0.4$, 24% of nodes and 60% of edges are missed. Note that 23% of LBR distributions in the Internet have a *GC* greater than 0.3, highlighting the limitations of the state-of-the-art MDA method, D-Miner, in discovering complete topologies.

3.2 Continuous Monitoring

Unlike prior research on Internet topology measurement, Internet topology monitoring aims to deliver continuous visibility

into the Internet's structural dynamics, which is critical for promptly detecting anomalies, and diagnosing failures.

A straightforward solution is to leverage existing measurement approaches to collect snapshots over time for Internet topology monitoring. However, three key limitations arise: 1) heavy network burden: the existing approaches generate large volumes of probing traffic to construct each topology snapshot, imposing a significant load on the Internet; 2) low probing efficiency: many Internet paths are relatively stable, and repeatedly verifying their existence with randomized flow identifiers leads to wasted probes; 3)lack of temporal context: because snapshots only capture isolated states, existing approaches cannot effectively distinguish whether a topology change is part of normal evolution or the result of an anomalous event.

To address these challenges, we propose estimating LBR distributions together with their temporal evolution. This approach brings three key benefits. First, it provides statistical guarantees on the completeness of measured topologies. Second, it enables more efficient probing by focusing resources on paths that are likely to have changed while avoiding redundant measurements on stable paths. Third, it facilitates anomaly detection: when the observed distributions deviate significantly from their predicted trends, the system can flag these deviations as unexpected changes, potentially signaling underlying network events.

4 Design of BayWatch

As a novel system for continuous multipath topology discovery, BayWatch utilizes a Dynamic Bayesian network to model the time-varying LBR distributions. In this section, we will first explain the basic idea of BayWatch, then demonstrate how to estimate the practical LBR distribution rather than simply assuming a uniform one, and finally describe how the time-varying transition probabilities are used to predict LBR distributions.

4.1 Basic Idea

IP networks adopt a hop-by-hop forwarding mechanism, *i.e.*, each router forwards packets independently based on its local routing table and the packet header fields, without considering the previous hops. This makes the forwarding decision dependent only on the current router's state, following the first-order Markov property [31, 34]. Furthermore, we hypothesize that the evolution of routing policies over time also adheres to a first-order temporal Markov property, where the network state at time t depends only on its state at time $t-1$.

Together, these two dimensions of spatial and temporal Markov properties make Dynamic Bayesian Networks (DBNs). In our model, nodes represent router interfaces and edges capture conditional forwarding probabilities. Temporal dependencies are expressed by connecting corresponding nodes across time slices. With this structure, estimating current LBR distributions becomes a parameter estimation problem in a Bayesian network, while predicting the most recent distribution becomes a temporal prediction problem in a DBN.

First-order markov hypothesis test for packet forwarding. To verify whether IP forwarding obeys a first-order Markov property, we use parametric hypothesis testing. We collect and organize a substantial routing dataset consisting of 300,000 state sequences, each state represents an IP address X_t at the t-th hop of a path. Then, we construct two models: a first-order Markov model M_1, which assumes that the next forwarding address k depends only on the current address j; and a more complex model, such as a second-order Markov model, which assumes that the next forwarding address k depends on both the current address j and the preceding address i.

$$M_1 : \hat{p_{jk}} = \frac{N_{jk}}{\sum_{k'} N_{jk'}} \tag{2}$$

where N_{jk} is the number of paths where $X_t = j$ and $X_{t+1} = k$.

$$M_2 : \hat{p_{ijk}} = \frac{N_{ijk}}{\sum_{k'} N_{ijk'}} \tag{3}$$

where N_{ijk} is the number of paths where $X_{t-1} = i$, $X_t = j$ and $X_{t+1} = k$.

Next, we calculate the goodness-of-fit between the two models and the actual data, that is, the likelihood, and use the likelihood ratio test to compare the merits of the two models.

$$L_1 = \sum_{j,k} N_{jk} log \hat{p_j} k \tag{4}$$

$$L_2 = \sum_{i,j,k} N_{ijk} log \hat{p_{ijk}} \tag{5}$$

After calculating the likelihoods of the two models, we assess the difference in how well the two models fit the data using a likelihood ratio test. This test relies on the likelihood ratio statistic Λ, which quantifies the ratio of the likelihoods of the two models, thereby measuring the degree of difference between the models.

$$\Lambda = -2(L_1 - L_2) \sim \chi^2 \tag{6}$$

To test statistical significance, we compute a p-value: the probability of obtaining a likelihood-ratio statistic at least as extreme as the observed one, assuming the null hypothesis—IP forwarding follows a first-order Markov process—is true. Under the null, the statistic is approximately distributed. Comparing the calculated p-value with $\alpha = 0.05$, we can determine whether to reject the null hypothesis. The p-value we obtained is 0.95, which is much higher than the significance level. This indicates that the topological dynamics satisfy the first-order Markov property.

First-order markov hypothesis test for topology dynamics. To empirically validate the first-order Markov property of

topological evolution, we collect 50 consecutive topology snapshots for each node and its successor found in our path samples.

During the data processing phase, we first align the LBR distribution presented by each node in every snapshot using the method described in Section 4.3. We then construct these into a discrete time series. Subsequently, we apply a rigorous hypothesis test to these series using the same Likelihood Ratio Test method as detailed above. This allows us to quantitatively assess whether these time series satisfy the assumption of a first-order Markov process, meaning the LBR distribution at the current time depends only on the state from the previous moment, not on earlier historical states. The p-value we obtain is 0.68, which is far greater than the significance level. This, similar to our findings for packet forwarding, indicates a high probability that we can accept the null hypothesis that topological dynamics satisfy the first-order Markov property.

Together with the forwarding test above, this result further supports the validity of modeling network dynamics using a DBN.

4.2 Parameter Estimation for LBR Distributions

To model the forwarding process using a Bayesian network, BayWatch organizes nodes with the same TTL from the source s into one layer. Each node in one layer is fully connected to the nodes in the next layer, including those without successors. This allows us to represent the LBR distribution as a transition probability matrix between adjacent layers [2]. Thus, estimating the LBR distribution becomes a parameter estimation problem in Bayesian networks with latent variables. Specifically, we consider an L-layer Bayesian network where the i-th layer contains X_i nodes ($i \in [1,L]$)[3]. After sending x probes with random flow identifiers to an IP prefix, we obtain an observed sequence $[y_1,y_2,...,y_x]$ in the last layer, where each observed value y_i corresponds to one of the nodes in the last layer. Our objective is to estimate three sets of parameters: (1) the initial distribution over the first-layer nodes, (2) the transition probabilities between adjacent layers, and (3) the observation probabilities that map the last-layer nodes to the observed values.

For efficiency, we apply variational inference [5, 25], which approximates the true posterior distribution with a more tractable one by maximizing the Evidence Lower Bound (ELBO). Let $Z^{(1)},Z^{(2)},...,Z^{(L)}$ denote the latent variables corresponding to the nodes in each layer. Our goal is to infer the posterior distribution $P(Z^{(1)},Z^{(2)},...,Z^{(L)}|y_1,y_2,...,y_x)$. However, this inference is generally intractable due to the computational complexity of summing over all possible configurations of the latent variables. To make this tractable, we introduce a variational distribution $Q(Z^{(1)},Z^{(2)},...,Z^{(L)})$, often assumed to factorize as [41]:

$$Q(Z^{(1)},Z^{(2)},...,Z^{(L)}) = \prod_{i=1}^{L} Q(Z^{(i)}) \tag{7}$$

We maximize ELBO to bring Q closer to the true posterior distribution. The ELBO is given by:

$$\text{ELBO} = \mathbf{E}_{Q(Z)}[\log P(y,Z)] - \mathbf{E}_{Q(Z)}[\log Q(Z)] \tag{8}$$

Firstly, we initialize the parameters. For each round, we construct a L-layer Bayesian network based on the topology obtained in the previous round and perform hypothesis testing. If we have discovered k nodes at hop i, we assume there are $k+1$ nodes at this hop, setting $X_i = k+1$. The initial probability distribution $P(Z^{(i)})$ is initialized using the softmax function, which converts the frequency of occurrence of each node from the topology obtained in the previous round into a probability distribution while smoothing to avoid zero values. Similarly, the transition probability $P(Z^{(i+1)}|Z^{(i)})$ is initialized based on the frequency of edges observed between the nodes.

Secondly, we begin iterating the Expectation (E) and Maximization (M) steps to maximize the ELBO. In the E-step, we hold the model parameters constant and update each $Q(Z^{(i)})$ ($i \in [1,L]$). According to Bayes' theorem, the posterior distribution is given by

$$Q(Z_j^{(i)}) \propto \exp(\mathbf{E}_{Q(\{Z^{(i)}\}-\{Z_j^{(i)}\})}[\log P(Z_j^{(i)},Y)]) \tag{9}$$

After updating all $Q(Z)$, we proceed to the M-step where we maximize the ELBO and then use gradient descent [13] to update the parameters. After that, the update rule for the initial probabilities of the first layer is derived from the expected value of $Q(Z_i^{(1)})$:

$$P(Z_i^{(1)}) = \frac{\mathbf{E}_{Q(Z_i^{(1)})}[Z_i^{(1)}]}{\sum_i \mathbf{E}_{Q(Z_i^{(1)})}[Z_i^{(1)}]} \tag{10}$$

In updating the transition probabilities $P(Z_j^{(t)}|Z_i^{(r)})$ between layer r and layer t, as well as the final observation probabilities, the rule is based on the expectation of $Q(Z_i^{(r)},Z_j^{(t)})$:

$$P(Z_j^{(t)}|Z_i^{(r)}) = \frac{\mathbf{E}_{Q(Z_i^{(r)},Z_j^{(t)})}[(Z_j^{(t)}|Z_i^{(r)})]}{\sum_j \mathbf{E}_{Q(Z_i^{(r)},Z_j^{(t)})}[(Z_j^{(t)}|Z_i^{(r)})]} \tag{11}$$

By substituting $Z_j^{(t)}$ with Y_j in the previous formula, we can update the observation probabilities from the last layer L to the observed values. The E-step and M-step are repeated iteratively until the ELBO converges [5, 25] or the predefined number of iterations is achieved.

4.3 Temporal Prediction of LBR Distributions

Beyond estimating current distributions, BayWatch predicts how LBRs evolve over time. A network snapshot is represented as a tensor $T = (M_1,...,M_{L-1})$, where each M_i is the transition matrix for layer i. $M_i[v_1][v_2]$ denotes the probability

[2] If there is no connection between two nodes, their transition probability is set to 0.

[3] Variables used in the probing process are summarized in Appendix C.

of forwarding from node v_1 at hop i to node v_2 at hop $i+1$. However, tensor shapes may be inconsistent since node sets may differ across snapshots. To this end, we propose a node alignment strategy: for each layer, we retain the k most frequently observed nodes across the past g snapshots and add an *others* node to aggregate all rare and newly discovered nodes. This produces tensors of uniform shape, enabling unified modeling. Hence, we can then use the Bayesian network to predict the probabilistic transition tensor of the $(g+1)$-th snapshot. For this, we introduce a variational distribution $Q(T_{g+1})$ for the $(g+1)$-th snapshot and again maximize ELBO:

$$\text{ELBO}=\mathbf{E}_Q[\log P(T_1,...,T_g,T_{g+1})]-\mathbf{E}_Q[\log Q(T_{g+1})] \quad (12)$$

Optimization yields an approximate posterior for model parameters. From this posterior, we derive predicted LBR distributions for nodes in the $(g+1)$-th snapshot, which then guide probing decisions and anomaly detection.

5 Workflow of BayWatch

BayWatch is a round-based probing approach, driven by the set of unresolved hops. To initiate this process, BayWatch requires an initial round to populate the set of unresolved hops. In this section, we first detail how to leverage prediction in Section 4.3 to predict the topology at the current moment which involves identifying the resolved and unresolved hops. We then illustrate how the LBR distribution, estimated in Section 4.2, is used to guide our round-based hypothesis testing and probing on this predicted topology.

5.1 Initializing the Algorithm

To initiate the hypothesis-testing algorithm for topology completeness, we must first obtain a baseline topology to serve as the foundation for the tests. BayWatch, provides two distinct modes for this purpose:

Bootstrap mode. This mode is used when no measurements have been performed before. Following the practice of prior work [39], we send n_1=6 probes with random flow identifiers to each hop towards each /24 prefix. This ensures that the probability of missing a second successor of a node is below 5%. Here, we adopt this initialization purely as a starting point for subsequent parameter estimation.

Predictive mode. In contrast, when historical time-varying LBR distribution has been obtained in previous measurements, BayWatch first predicts the current LBR distributions. Edges with non-zero probability are hypothesized to exist, while edges with zero probability are treated as absent. The initial probing round then focuses on verifying these predictions. For edges associated with frequently observed nodes, we reuse flow identifiers from prior snapshots to validate them efficiently. This approach yields a baseline topology at substantially lower cost, avoiding the heavy initialization of the bootstrap mode.

Note that both the bootstrap mode and the predictive mode rely on multiple rounds of observation to progressively approximate the true LBR distribution. The key difference lies in their starting point: the bootstrap mode begins without any prior knowledge, while the predictive mode leverages historical time-varying LBR distribution as prior knowledge, allowing it to start from a more accurate point and complete the approximation process much faster, significantly improving probing efficiency. However, the predictive mode includes an exception handling mechanism. If, during the probing process at a given hop, a new edge is discovered that has not appeared in any historical snapshots, then the statistical priors built from the historical data are invalidated. In this case, BayWatch must discard its prior knowledge for that specific hop and fall back to the bootstrap mode.

5.2 Optimizing Probes at Hop Level

Similar to MDA [8], BayWatch calculates the minimum number of probes needed to statistically guarantee discovery of all successors.

Let R_h denote the set of nodes at hop h, with $v \in R_h$ and $v' \in R_{h+1}$. The forwarding probability from node v to node v' is $M_h(vv')$. The minimum number of probes required to provide a statistical guarantee for discovering all edges between node v and its successors is given by:

$$m(v)=\max_{v' \in R_{h+1}} \left\lceil \frac{\log(1-\alpha)}{\log(1-M_h(vv'))} \right\rceil \quad (13)$$

Next, based on these equations about a node, we calculate the minimum number of probes required at each hop. For a hop h, we hypothesize the presence of an undiscovered node v_h^* at hop h, in addition to the already discovered nodes. Under the hypotheses, we proceed to estimate the parameters M_h and P_h, where P_h represents the distribution that probes with TTL h from s reach each node at h-th hop. For statistical guarantee of all edges, we must ensure the edges between each node $v \in R_h$ and its successors are discovered. Therefore, for a hop h, the minimum number of probes should be sent is given by:

$$n^h=\max\left(\max_{v \in R_{h-1}}\left(\left\lceil \frac{m(v)}{P_{h-1}(v)} \right\rceil\right), \max_{v \in R_h}\left(\left\lceil \frac{m(v)}{P_h(v)} \right\rceil\right)\right) \quad (14)$$

BayWatch employs a round-based probing method, which allows the number of probes required to meet a statistical guarantee to be accumulated across multiple rounds. Specifically, if t_h probes have already been sent to a hop h in previous rounds, BayWatch subtracts t_h from the minimum total number of probes required for that hop in the current round.

This reuse of probes is especially valuable in the bootstrapping round, where we validate an initial topology predicted from historical data. If no new links are observed, we can infer that earlier probes have already satisfied part of the hypothesis testing. In that case, BayWatch avoids re-probing those stable paths, significantly reducing overhead.

5.3 Round-based Probing

After the bootstrapping round, BayWatch calculates the number of probes required in a new round based on Equation (14). This process is repeated until the number of probes dispatched to all hops diminishes to zero, indicating that all nodes and edges have been discovered statistically.

Figure 4 illustrates how BayWatch models a 4-hop multipath topology into a 4-layer Bayesian network. First, after the bootstrapping round, we obtain the topology in Figure 4a, including 5 nodes and 4 edges. The probability distributions of probes from the source s reach nodes at the second and third hop are P_2 and P_3, respectively, and the transition probability from the second hop to the third hop is M_2.

$$P_2=[\frac{11}{20},\frac{1}{4},\frac{1}{5}],P_3=[\frac{29}{120},\frac{73}{120},\frac{3}{20}],M_2=\begin{bmatrix}\frac{1}{3} & \frac{1}{2} & \frac{1}{6}\\ \frac{1}{10} & \frac{8}{10} & \frac{1}{10}\\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6}\end{bmatrix}$$

We take the second hop as an example to introduce how to calculate the number of probes with 95% confidence. Hypothesize that there is a third node in both hops, named v_2^* and v_3^* respectively, and the nodes within the two hops are fully connected. Using the maximum likelihood estimation, since v_2^* and v_3^* have not appeared, their probabilities are expected be the minimum, estimated as $\frac{1}{5}$ and $\frac{3}{20}$.

Then, we take v_2^1 as an example to illustrate probes required to discover a node's successor edges. According to Equation (15), 5 probes are required to validate the existence of the edge between v_2^1 and v_3^2. Therefore, $\max\{5,8\}=8$ probes are required to validate all edges between v_2^1 and discovered nodes at TTL 3. Similarly, $\max\{29,2\}=29$ probes and $\max\{17,3\}=17$ probes are required to discover all successor edges of v_2^A and v_2^*, respectively. Therefore, $\max\{8\times\frac{20}{11},29\times\frac{4}{1}\}=116$ probes are required to discover successor edges of v_2^1 and v_2^A, and $17\times\frac{5}{1}=85$ probes are required to discover successor edges of v_2^*.

Assuming with these probes, BayWatch validates the existence of v_2^*, denoting it as v_2^3. Besides, BayWatch discovers the edge between v_1^1 and v_2^3. However, v_3^* is not seen, suggesting that v_3^* does not exist statistically at TTL 3. Hence, the multipath topology is updated as the one in Figure 4b. Note that v_2^* is a new hypothetical node, distinct from that in Figure 4a. Besides, P_2 and M_2 are updated as follow:

$$P_2=[\frac{9}{20},\frac{1}{4},\frac{1}{5},\frac{1}{10}],M_2=\begin{bmatrix}\frac{5}{12} & \frac{7}{12}\\ \frac{3}{20} & \frac{17}{20}\\ \frac{1}{3} & \frac{2}{3}\\ \frac{1}{2} & \frac{1}{2}\end{bmatrix}$$

Similar to ROUND I, BayWatch calculates that it needs to send 50 probes to validate the existence of the successor edges of v_2^*. Based on the updated LBR distribution, 76 probes are required to discover successor edges of v_2^1, v_2^A, and v_2^3, fewer than the 85 probes calculated in the last round for discovering edges of v_2^3. Therefore, we need $\max\{0,76,29\}=76$ probes for TTL 2 in ROUND II.

Assuming v_2^* is not detected by these probes. BayWatch will send 50 probes in Round III to meet the statistical guarantees of edges between the second and third hop. This process discovers the edges between v_2^1 and v_3^2, as well as between v_2^3 and v_3^2. As a result, BayWatch will conclude with 95% confidence that the topology has been completely discovered, and is the one in Figure 4c. Overall, in the three rounds, BayWatch totally sends 110 probes to the second hop instead of 116+76+50=242, which is the number of probes required by D-Miner's implementation.

6 Validation on Ground Truth

To validate the topology discovery capability of BayWatch, we adopt *the same experiment settings in Section 3.1.2*, and first evaluate the accuracy of the estimated LBR distributions, and then demonstrate that BayWatch achieves at least 70% edge recall in nearly 90% of the topologies, while D-Miner only achieves 35%.

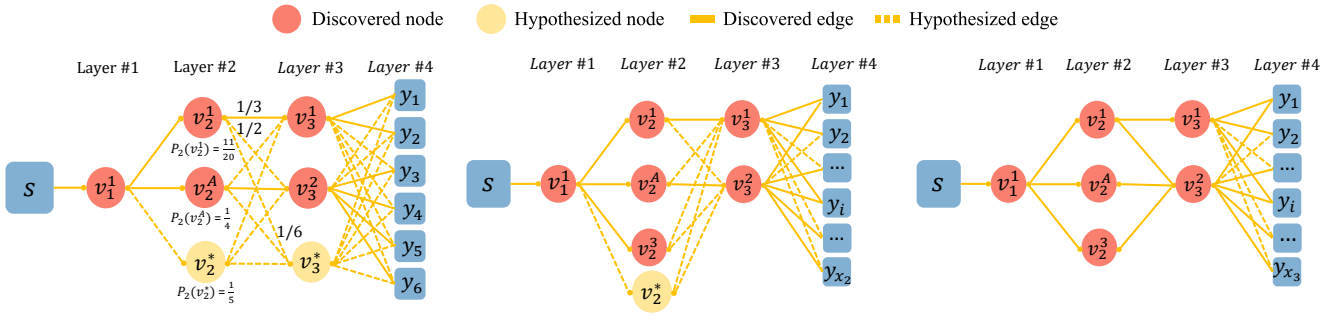
6.1 Accuracy of LBR Distribution Estimation

We employ the Jensen-Shannon Divergence (JSD) metric [42] to measure the similarity between the estimated and actual LBR distributions. JSD values range from 0 to 1, where values approaching 0 signify greater similarity. As the LBR distribution becomes more skewed, certain nodes may be missed, resulting in a dimensional discrepancy between the estimated and actual distributions. In such cases, we fill the missing dimensions in the estimated or actual distributions with zeros before calculating the JSD.

As shown in Figure 5, the average JSD between the estimated and actual LBR distributions is close to 0 when $GC\leq 0.5$, which applies to 89.4% of DNs (see Figure 2). As GC increases from 0.5 to 0.8, the number of undetected nodes gradually rises, causing a slow increase in JSD. Once GC exceeds 0.8, the JSD grows rapidly due to the huge number of undetected nodes, widening the gap between the estimated and actual distributions. Therefore, BayWatch accurately estimate the LBR distribution in most Internet cases, providing an important basis for reliable statistical guarantees.

6.2 Topology Completeness

The topology uncovered by BayWatch under the same experiment settings as D-Miner is shown in Figure 3. When $GC\leq 0.5$, BayWatch can discover 85.86% of the nodes, decreasing only 4.76% from $GC=0$, *i.e.*, uniform LBR distribution, to $GC=0.5$. Even at a GC of 0.9, BayWatch still discover 68.60% of the nodes, 21% more than D-Miner. For edges, BayWatch maintains almost the same edge recall when $GC\leq 0.3$, covering 77% of DNs on the Internet. While the number of missed edges increases as the GC grows beyond 0.3, BayWatch still outperforms D-Miner by 11.32% to 38.20%.



(a) Bayesian network in ROUND I (b) Bayesian network in ROUND II (c) Bayesian network in ROUND III

Figure 4: The evolution of a Bayesian network in three rounds. v_2^A corresponds to the set of anonymous nodes. v_2^* and v_3^* correspond to hypothetical nodes at the second and third hops, respectively. Numbers alongside the edges are the probability that a probe is forwarded from the predecessor node to the successor node.

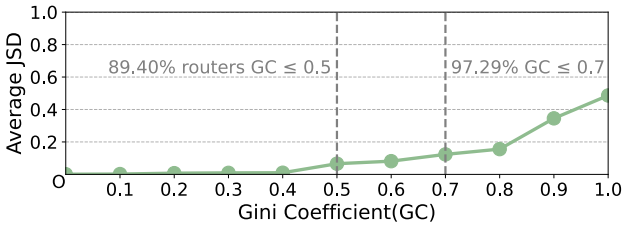


Figure 5: JSD between the estimated and actual distributions.

In summary, combined with the JSD between the estimated and actual distribution shown in Figure 5, the decline in recall can be attributed to inaccuracies in the estimated LBR distribution, highlighting the importance of LBR distribution for topology discovery.

7 Monitoring the Internet with BayWatch

In this section, we evaluate BayWatch's performance in Internet monitoring. For a single snapshot, BayWatch discovers 2.4× more nodes and 2.8× more links than D-Miner. When BayWatch is used for continuous monitoring, we show that BayWatch's predictive mode reduces probing overhead to 15.85% of the original cost with a completeness loss of less than 1%. Remarkably, it achieves 2.8× the completeness of D-Miner at just 40% of D-Miner's per-snapshot cost.

7.1 Experiment Setup

Similar to D-Miner, we use the IPv4 /24 prefix as the destination. The /24 prefix length is selected because it is commonly considered the longest acceptable prefix in BGP routing [37]. To obtain an Internet-scale topology, we probe all IPv4 /24 prefixes, excluding private and reserved ones [16], totaling 14,461,948. We use a virtual private server (VPS) in Beijing as the scanner and conduct independent measurements, capturing 13 snapshots from July 1 to September 6, 2025. We also use 10 geographically diverse VPSes across Asia, Europe and the Americas to capture 3 snapshots to mitigate

Table 1: Snapshots of Internet topology by bootstrap mode

Algorithm	Snapshot 1	Snapshot 2	Snapshot 3
Nodes			
RIPE Atlas	673,335	693,974	514,037
Yarrp	604,943	684,947	633,778
D-Miner	959,610	1,086,768	1,052,121
BayWatch	**2,159,102**	**2,608,245**	**1,892,796**
Links			
RIPE Atlas	1,973,111	1,971,225	1,176,221
Yarrp	1,324,566	992,123	946,389
D-Miner	2,275,933	2,934,063	2,031,384
BayWatch	**7,163,165**	**8,692,028**	**6,518,245**

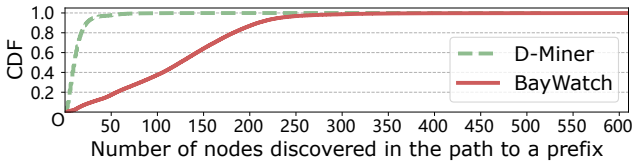
location bias (Details in Table 5). To minimize interference to the Internet, we limit the probing rate to 200,000 packets per second. We use UDP-based probes, as they are more effective at discovering links compared to other protocols [30, 39].

Our approach to varying flow identifiers differs from the existing method [39], which modifies only the destination address and port numbers. This method may miss LB mechanism that solely hashes based on port number or DSCP field [4]. To ensure that all types of LB mechanisms can be captured, we modify the destination address, source port, destination port, and DSCP field simultaneously when varying flow identifiers. Since the existing method correspond to a restricted case of our strategy, we run both D-Miner and BayWatch under this unified flow identifier variation scheme.

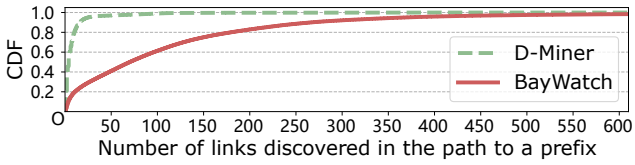
7.2 Topology Discovered in Single Snapshot

We treat D-Miner as the primary baseline [39], and also compare BayWatch with two additional baselines: RIPE Atlas [33] and Yarrp [9]. Table 1 shows the Internet topology snapshots during continuous monitoring discovered by BayWatch and the baselines. The three snapshots shown here are generated using the bootstrap more, probing from scratch without any DBN-based initial topology prediction.

Compared to the state-of-the-art method of MDA, BayWatch discovers 2.44 ∼ 2.67× nodes and 2.82 ∼ 3.37× links



(a) CDF of the number of nodes per prefix



(b) CDF of the number of links per prefix

Figure 6: CDF of nodes and links discovered per prefix.

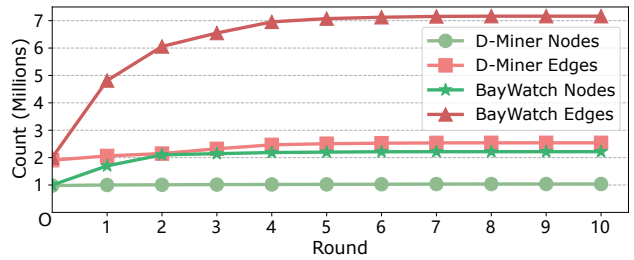


Figure 7: Cumulative number of nodes/links discovered in each round.

of D-Miner in the first three snapshots. Specifically, BayWatch discovers over 2 million nodes and 7 million links in each snapshot. In contrast, D-Miner discovers only around 1 million nodes and fewer than 2.6 million links in each snapshot, consistent with the single snapshot reported in the original work of D-Miner [39]. The significant difference in topology scale between BayWatch and D-Miner aligns with the recall gap shown in Figure 3. Interestingly, *this is the first time to uncover the Internet topology obscured by uneven LBR distributions, revealing millions of previously hidden nodes and links.* We also run Yarrp using our VPSes, and compare BayWatch with its result and the data from RIPE Atlas [33] that has 11,711 vantage points. We find that multipath topology detection algorithms discover more nodes and links than these single path tracing tools.

Next, we analyze how BayWatch discovers more nodes and links than prior work by focusing on three key aspects:

(1) **Topology discovered for a single prefix.** Figure 6 shows the number of nodes and links discovered by BayWatch and D-Miner for a single prefix. D-Miner discovers fewer than 40 nodes and links for 96% of prefixes. In contrast, BayWatch discovers fewer than 40 nodes for only 13% of prefixes and fewer than 40 edges for 36%. Additionally, BayWatch discovers more than 160 links for 23% of prefixes, while D-Miner does so for only 0.35%. We verified some results containing routers with over 80 successors with a large ISP, which confirmed the existence of these links. D-Miner, however, discovers fewer than 40% of these links.

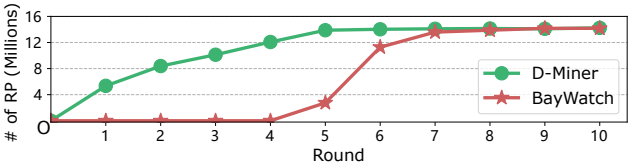


Figure 8: Cumulative number of resolved prefixes (RP) in each round.

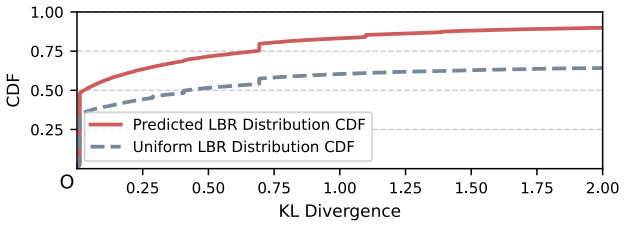


Figure 9: CDF of KL divergence between predicted LBR distributions and ground truth.

(2) **New nodes and links discovered in each round.** Figure 7 compares the cumulative discoveries per round for BayWatch and D-Miner. D-Miner discovers most nodes and links in the first round, adding only 20,499/218,236 in subsequent rounds. In contrast, BayWatch continues to discover many new nodes and links across several rounds, adding 72,482/615,493 in total. This difference stems from D-Miner's uniform distribution assumption, which results in fewer probes. In contrast, BayWatch estimates the LBR distribution and sends enough probes for statistical guarantee, leading to the discovery of new nodes and links.

(3) **Prefixes resolved in each round.** A prefix that does not contribute any new nodes or edges is considered *resolved*. Figure 8 shows the number of resolved prefixes in each round for both BayWatch and D-Miner. D-Miner resolves about 50% of the prefixes in the first round, indicating that it discovers no new nodes or links for these prefixes in this round. In contrast, BayWatch continues to discover new nodes and links across multiple rounds, up to the fifth round. This difference arises because D-Miner, due to its assumption of uniform distribution, sends fewer probes than BayWatch when targeting the same prefixes. As a result, D-Miner marks the prefixes as resolved too early to discover additional nodes and edges.

7.3 Topology Discovered in Continuous Monitoring

Using the first three snapshots as the initial historical data, our evaluation on continuous monitoring proceeds from the fourth snapshot onward. For each new snapshot, we run two versions of our BayWatch framework in parallel: a predictive mode and a bootstrap mode. The result from the bootstrap mode is used as the ground truth to check two aspects of the initial topology generated by the predictive mode: the accuracy of its LBR distributions and its completeness.

Table 2: Snapshots of Internet topology by predictive mode

Snapshot	Nodes(P)	Nodes(B)	Links(P)	Links(B)
4	2,448,310	2,450,613	7,322,468	7,322,468
5	2,191,159	2,195,362	7,250,477	7,253,371
6	2,561,608	2,567,024	7,808,499	7,813,153

7.3.1 Accuracy of Predicted LBR Distributions

The evaluation results demonstrate that BayWatch achieves strong predictive performance. Specifically, when determining the existence of a link (*i.e.*, whether an LBR distribution is nonzero), BayWatch attains an average accuracy of 80.05% across all snapshots. To further evaluate the quality of predicted LBR distributions, we conduct an in-depth analysis on a randomly selected snapshot by computing the KL divergence between the predicted distributions and the corresponding ground truth for all nodes.

The overall effectiveness of the predictive mode is illustrated in Figure 9. About half of the predicted LBR distributions exactly match the ground truth (*i.e.*, KL divergence of 0), while nearly 80% achieve a KL divergence of less than 1. Moreover, compared with a baseline that assumes a uniform distribution, BayWatch improves the accuracy of LBR distribution by about 20%. The more accurate prediction of LBR distributions yields two key advantages: first, it can provide high-quality initial values for subsequent parameter estimation in the DBN, thereby accelerating the convergence of LBR distribution estimation. Second, it can serve as a stable baseline for network behavior, enabling the detection of anomalies as shown in Section 7.7.

7.3.2 Topology Discovered by Predictive Mode

Table 2 compares the number of discovered nodes and edges between our predictive mode and bootstrap mode across three snapshots, and the rest can be found in Appendix E. By initiating hypothesis testing from a predicted topology, the predictive mode of BayWatch achieves a final result with a completeness loss of less than 1%. As shown in Section 7.5, the predictive mode reduces the number of probes sent by the scanner by approximately 84%. We believe this trade-off makes BayWatch a more cost-effective solution for continuous monitoring.

7.4 Validation on ISP Networks

We validate the links discovered in a large ISP network. According to the ISP's IP address database, there are 108,752 links where both endpoints belong to the ISP's network. The ISP reported that 97.12% of these links were validated within its network, and the rest were gone due to the dynamic nature of its network. Note that similar to D-Miner, BayWatch adopts the same encoding strategy as Paris traceroute [6]. That is, for a given destination IP address, certain fields remain fixed for all TTL probe values to guarantee that these probes take the same forwarding path. Hence, we believe that *BayWatch can avoid false links at adjacent hops*.

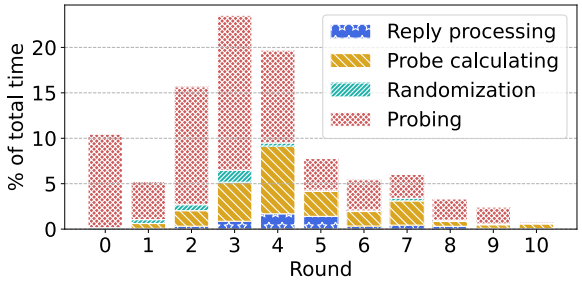


Figure 10: Time spent on different phases in each round.

7.5 Probe Overhead

In bootstrap mode, BayWatch sends 11.35 billion probes to discover 2.5 million nodes and 7.6 million links, while D-Miner sends 4.32 billion probes to discover 0.96 million nodes and 2.2 million links. That is, BayWatch can discover 0.67 links with 1,000 probes, 1.3× that of D-Miner.

Our analysis of these snapshots reveals that only 8.06% of hops rehash their successors across different snapshots. This provides a crucial insight: for the vast majority of hops, sending probes with flow identifiers previously used in previous snapshots will not yield new nodes. Based on this insight, BayWatch reuses probes from previous snapshots to provide a partial statistical guarantee for these hops, instead of random rediscovering all paths from scratch. This mechanism drastically cuts down on redundant probing. That is, the predictive mode reduces the number of probes to 1.8 billion, only 15.8% of the bootstrap mode, while incurring a negligible loss in topology completeness.

Our snapshots reveal that in some cases, the number of probes in the predictive mode surged to 2.3 billion. Our further analysis attributes this to novel topological change patterns absent from previous snapshots, making BayWatch fall back to the bootstrap mode. In particular, we observe that certain operator nodes maintain a stable set of next-hop addresses over a period, with a small fluctuation in the number of next hops across different snapshots. However, after some time, this stable set is entirely replaced by a new set of IP addresses, whose size also stabilizes at a different value.

7.6 Time Cost

Using bootstrap mode, it takes BayWatch 52 hours to complete an Internet-wide measurement. In contrast, the time cost is reduced to 7.82 hours by using the predictive mode, 6.6× reduction from the bootstrap mode. This massive reduction in time allows for more frequent probing, enabling to capture dynamic changes in Internet topology that were previously ignored due to the high time cost of measurement.

We further break down the time spent on four phases in each round in the bootstrap mode, including reply processing, probe calculation, randomization, and probing. Specifically, replies are processed by filtering out fully discovered targets and hops. Probe calculation involves estimating LBR distributions and determining the number of probes needed for the current round, which is central to BayWatch's design. During the randomization phase, we adjust flow identifiers and randomize the probe sending order. Finally, in the probing phase, BayWatch sends out the probes and waits for responses.

Figure 10 shows a stacked bar chart of the time consumed by these phases in each round. The initial round takes slightly longer, *i.e.*, 5.5 hours, because BayWatch sends probes to all 2∼32 hops for each target. However, most paths have a length of around 10 hops, about half of these probes do not reach the expected hop, but respond to the targets in advance *i.e.*, without ICMP exceeded responses. Hops beyond the targets are excluded in the subsequent rounds, reducing the probing load[4]. As more paths are discovered, BayWatch discovers over 90% of the links in rounds 2, 3, and 4, which leads to an increase in the duration of each phase before the fifth round. Starting from the fifth round, the focus shifts mainly to verifying the links discovered in the first four rounds, and as new discoveries decrease, probing begins to converge.

7.7 Case Study on Anomaly Detection

During continuous Internet monitoring, BayWatch tracks nodes whose LBR prediction accuracy persistently deviates from expected behavior, enabling the detection of anomalous network events. On August 15, 2025, we observed a sharp surge in rehashing nodes within a subset of Zayo's network, where routers began returning different next-hop addresses for identical flow identifiers, deviating from their usual stable forwarding behavior. Cross-referencing external reports confirmed that this event coincided with a widely reported service outage involving Zayo, demonstrating that BayWatch can effectively identify topology anomalies and link them to real-world service disruptions. Detailed analysis of this incident is provided in Appendix H.

8 Related Work

Limitations of single path measurement. The traditional traceroute tool [7, 24] increases TTL to get router responses, attempting to discover a single path from source to destination. However, load balancing mechanisms can prevent it from detecting all potential paths, as responses may come from multiple routes. Partial topologies can lead to biased conclusions about the network's properties [3, 15, 43]. Yarrp improves efficiency by encoding state information into packets [9, 10, 44], enabling parallel probing to avoid network overload. While it can effectively probe all IPv4 /24 prefixes, Yarrp still has limitations in handling multipath networks.

Load balancing measurement and characterization. MDA introduces a multipath discovery approach, providing the first detailed analysis of diamonds in networks [8]. However, with the rise of complex technologies like SDN and advanced traffic engineering, load balancing mechanisms have grown increasingly intricate. MCA categorizes these mechanisms [4], but its conclusions are limited by probing only around 10,000 IPs, making it insufficient for the complexity of the Internet. In contrast, D-Miner, with its Internet-wide approach, reveals the expanded size of diamonds [39]. However, its reliance on uniform load balancing assumptions leads to the omission of more complex diamond structures.

Router dynamics discovery. Paxson's study [32] finds that Internet routes remain globally stable over time, though it does not consider load balancing. Nearly a decade ago, Cunha *et al.* reassessed Paxson's findings by including load balancing and confirmed their validity [18], noting that load balancing remapping was rare. Five years ago, Vermeulen *et al.* used D-Miner to study a large-scale multipath topology and discovered that over half of route changes were due to load balancing remapping [39]. Sibyl [17] predicts query-relevant Internet paths from historical measurements rather than modeling routing dynamics. However, incomplete discovered topology data may have led to an overestimation of Internet dynamics.

9 Conclusion

In this paper, we present BayWatch, a practical Internet-scale topology monitoring system that overcomes two major limitations of prior work: 1) incompleteness caused by the simplified assumption of uniform LBR distributions, and 2) low probing efficiency due to the lack of temporal awareness. By modeling packet forwarding with a DBN, BayWatch estimates true LBR distributions and predicts their temporal evolution. Internet-wide experiments show that BayWatch uncovers millions of nodes and links missed by D-Miner, while its predictive mode reduces time cost by 6.6×. Moreover, BayWatch demonstrates its capability in supporting anomaly detection through identification of deviations from expected topology dynamics.

Acknowledgments

We sincerely thank our shepherd Prof. Ramesh Govindan and all anonymous reviewers for their constructive comments. This work was supported by National Key R&D Program of China (2022YFB3105000), the Beijing Outstanding Young Scientist Program (No. JWZQ20240101008) and NSFC under Grant 62502472. This work was also supported by Zhongguancun Laboratory. Shuai Wang and Zhaoteng Yan are the corresponding authors.

[4] In practice, 2 hops beyond the targets are still included to avoid missing nodes in a longer path, which is similar with D-Miner.

References

[1] Measurement lab (m-lab). https://www.measurementlab.net. Accessed: September 1, 2024.

[2] Planetlab europe. https://www.planet-lab.eu. Accessed: September 1, 2024.

[3] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. Anatomy of a large european ixp. In *Proceedings of the ACM SIGCOMM 2012*, pages 163–174, 2012.

[4] Rafael Almeida, Renata Teixeira, Darryl Veitch, Christophe Diot, et al. Classification of load balancing in the internet. In *IEEE INFOCOM 2020*, pages 1987–1996. IEEE, 2020.

[5] Hagai Attias. A variational baysian framework for graphical models. *Advances in neural information processing systems*, 12, 1999.

[6] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 153–158, 2006.

[7] Brice Augustin et al. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 153–158. ACM, 2006.

[8] Brice Augustin, Timur Friedman, and Renata Teixeira. Measuring load-balanced paths in the internet. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 149–160, 2007.

[9] Robert Beverly. Yarrp'ing the internet: Randomized high-speed active topology discovery. In *Proceedings of the 2016 Internet Measurement Conference*, pages 413–420, 2016.

[10] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. In the ip of the beholder: Strategies for active ipv6 topology discovery. In *Proceedings of the Internet Measurement Conference 2018*, pages 308–321, 2018.

[11] Kenneth L Calvert, Matthew B Doar, and Ellen W Zegura. Modeling internet topology. *IEEE Communications magazine*, 35(6):160–163, 1997.

[12] Catchpoint. Learnings from servicenow's proactive response to a network breakdown. https://www.catchpoint.com/blog/learnings-from-servicenows-proactive-\response-to-a-network-breakdown, 2025. Accessed: 2025-09-17.

[13] Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[14] Cisco. Thousandeyes announces real-time internet outage detection. https://www.thousandeyes.com/press-releases/internet-outage-detection, 2016. Accessed: 2025-09-17.

[15] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. Internet mapping: from art to science. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 205–211. IEEE, 2009.

[16] M. Cotton, L. Vegoda, and R. Bonica. Special-purpose ip address registries. RFC 6890, April 2013.

[17] Ítalo Cunha, Pietro Marchetta, Matt Calder, Yi-Ching Chiu, Bruno VA Machado, Antonio Pescapè, Vasileios Giotsas, Harsha V Madhyastha, and Ethan Katz-Bassett. Sibyl: a practical internet route oracle. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 325–344, 2016.

[18] Ítalo Cunha, Renata Teixeira, Darryl Veitch, and Christophe Diot. Dtrack: A system to predict and track internet path changes. *IEEE/ACM Transactions on Networking*, 22(4):1025–1038, 2013.

[19] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *ACM SIGCOMM computer communication review*, 29(4):251–262, 1999.

[20] Dioptra Group. Diamond-miner: An efficient probe scheduling and network topology discovery tool. https://github.com/dioptra-io/diamond-miner, 2024. Accessed: August 19, 2024.

[21] Joe Hasell. Measuring inequality: what is the gini coefficient? https://ourworldindata.org/what-is-the-gini-coefficient, 2023.

[22] Huawei. Ne40e-f v800r022c00spc600 feature description. https://support.huawei.com/enterprise/en/doc/EDOC1100278527/914792b5, 2022. Accessed: 2024-09-12.

[23] Huawei. Ar500, ar510, ar531, ar550, ar1500, and ar2500 v200r010 cli-based configuration guide - mpls. https://support.huawei.com/enterprise/en/doc/EDOC1100034233/622a51a3, 2024. Accessed: 2024-09-12.

[24] Van Jacobson. traceroute. In *Proc. USENIX Winter*, pages 1–10, 1989.

[25] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

[26] Piotr Jurkiewicz. Topohub: A repository of reference gabriel graph and real-world topologies for networking research. *SoftwareX*, 24:101540, 2023.

[27] Nanxi Kang, Monia Ghobadi, John Reumann, Alexander Shraer, and Jennifer Rexford. Efficient traffic splitting on commodity switches. In *Proceedings of the 11th ACM CoNEXT*, pages 1–13, 2015.

[28] KI3 Team. KI3 IP Topology. [Online]. Available: https://ki3.org.cn/#/dataset?name=ip_topology, 2026.

[29] LogicMonitor. Topology mapping. https://www.logicmonitor.com/support/forecasting/\topology-mapping/topology-mapping-overview, 2025. Accessed: 2025-08-19.

[30] Matthew Luckie, Young Hyun, and Bradley Huffaker. Traceroute probe method and forward ip path inference. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 311–324, 2008.

[31] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. John Wiley & Sons, New York, 1989.

[32] Vern Paxson. End-to-end internet packet dynamics. In *Proceedings of the ACM SIGCOMM*, pages 139–152, 1997.

[33] RIPE NCC. RIPE Atlas Measurement Data: ID 5051. https://atlas.ripe.net/measurements/5051/, 2024. Accessed: 2024-08-19.

[34] S. M. Ross. *Introduction to Probability Models*. Academic Press, San Diego, 10th edition, 2010.

[35] SolarWinds. Network topology - automate topology mapping. https://www.solarwinds.com/network-topology-mapper/use-cases/network-topology, 2019. Accessed: 2025-09-17.

[36] Raffaele Sommese. anycast-census. https://github.com/anycast-census/anycast-census, 2025. Accessed: 2025-09-17.

[37] Stephen Strowes. Visibility of prefix lengths in ipv4 and ipv6. https://labs.ripe.net/Members/stephen_strowes/visibility-\of-prefix-lengths-in-ipv4-and-ipv6, 2023. Accessed: 2024-08-21.

[38] Darryl Veitch, Brice Augustin, Renata Teixeira, and Timur Friedman. Failure control in multipath route tracing. In *IEEE INFOCOM 2009*, pages 1395–1403. IEEE, 2009.

[39] Kevin Vermeulen, Justin P Rohrer, Robert Beverly, Olivier Fourmaux, and Timur Friedman. Diamond-miner: Comprehensive discovery of the internet's topology diamonds. In *NSDI*, pages 479–493, 2020.

[40] Kevin Vermeulen, Stephen D Strowes, Olivier Fourmaux, and Timur Friedman. Multilevel mda-lite paris traceroute. In *Proceedings of the IMC 2018*, pages 29–42, 2018.

[41] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[42] Wikipedia. Jensen–shannon divergence. https://en.wikipedia.org/wiki/Jensen%E2%80%93Shannon_divergence, 2024. Accessed: 2024-09-12.

[43] Walter Willinger, David Alderson, and John C Doyle. Mathematics and the internet: A source of enormous confusion and great potential. *Notices of the AMS*, 56(5):586–599, 2009.

[44] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, and et. al. Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *Proceedings of ACM SIGCOMM*, pages 432–445, 2017.

[45] Junlan Zhou, Malveeka Tewari, Min Zhu, Abdul Kabbani, Leon Poutievski, Arjun Singh, and Amin Vahdat. Wcmp: Weighted cost multipathing for improved fairness in data centers. In *Proceedings of the Ninth European Conference on Computer Systems*, pages 1–14, 2014.

Appendix

A Ethics

Similar to previous research that uses active measurements, this work involves probing the Internet and takes precautions to minimize potential risks associated with such probing. To avoid disrupting the Internet, particularly to prevent overloading remote networks, we limit each VPS to a probing rate of 200,000 packets per second and randomize the probing sequence. Additionally, we set a PTR record for the IP addresses of each VPS, allowing network operators to contact us if they wish to opt out of the scanning.

No other aspects of this work raise ethical issues.

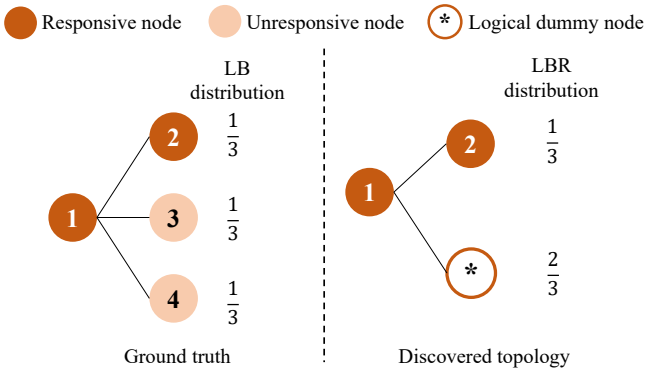


Figure 11: Impact of ICMP non-responsive on D-Miner.

B Causes of Uneven LBR Distribution

Three primary factors cause the uneven LBR distributions in the Internet.

(1) WCMP Deployment: We observe a common pattern of $\{0.25,0.25,0.125,0.125,0.125,0.125\}$ for distributions with a *GC* of 0.17, where some successors receive twice the packets of others. This likely results from WCMP configuration [22, 45], which adjusts traffic distribution based on link bandwidth differences to improve bandwidth utilization. For instance, assigning weights of 2:2:1:1:1:1 to 6 outgoing links leads to the observed distribution.

(2) MPLS Tunnel: We observe extreme distributions, such as $\{0.98,0.02\}$ (GC=0.96), likely caused by MPLS, as indicated by MPLS labels in ICMP error responses. MPLS is a scalable solution to implement traffic engineering in large-scale backbone networks [23, 27]. As an overlay model, MPLS establishes tunnels over the physical topology to route traffic. This may result in uneven flow distribution across physical routers.

(3) ICMP Non-Response: The measurement results show that 90% of paths contain more than 5 hops that do not respond to ICMP requests. The unresponsiveness makes it hard to determine the number of unresponsive nodes at a hop. A common approach is to abstract them into a *logical dummy node*, and regard the non-response as a special *dummy response*, whose probability is the sum of the probability of unresponsive successors. Therefore, even if the LB distribution among all successors is even, the LBR distribution may be uneven due to unresponsive successors. As shown in Figure 11, node 1 forwards packets equally to 3 successors. When the sent probes are forwarded to node 2, node 3 and node 4, we will observe responses from node 2 and some non-responses. If we assume these non-responses come from a logical dummy successor [5], including node 3 and node 4, its probability is twice that of the responsive node 2, *i.e.*, $\frac{2}{3}$.

[5]Note that ignoring unresponsive successors will further reduce the minimum expected number of probes from n_2=11 to n_1=6, failing to provide statistical guarantee.

Table 3: Summary of variables in probing process.

Variable	Description
v	A node representing an IP interface
h	TTL h (or the h-th hop)
v_h^*	A hypothesized node at TTL h
R_h	The set of nodes at TTL h
P_h	Probability distribution of probes from s reaching nodes at TTL h
M_h	The matrix of probabilities that probes from TTL h reach nodes at TTL h+1
$n(v)$	The minimum number of probes to statistically discover node v
$m(v)$	The minimum number of probes to statistically discover all edges between node v and its successors
t_h	The total number of probes that have been sent to TTL h in previous rounds
L	The number of layers of the Bayesian network
X_i	The number of nodes in the i-th layer of the Bayesian network
Y	The value of observations
x	The length of observation sequence
$Z^{(i)}$	The latent variables corresponding to the nodes in the i-th layer

C Variable Summary

Variables used in Section 4 are summarized in table 3.

D Proof

Proof. Given $v' \in R_{h+1}$, let $m(vv')$ denote the minimum number of probes required to guarantee that the probability of discovering the edge vv' is greater than α. Then, $m(vv')$ should satisfy the condition $1-(1-M_h(vv'))^{m(vv')} \geq \alpha$. Therefore, $m(vv')$ can be calculated as follows:

$$m(vv') = \left\lceil \frac{\log(1-\alpha)}{\log(1-M_h(vv'))} \right\rceil \tag{15}$$

Therefore, to ensure that every successor edge of v meets statistical guarantee, the minimum number of probes required is $m(v) = \max_{v' \in R_{h+1}} m(vv')$.

E Snapshots

The snapshots introduced in Section 7.1 are presented in Table 4.

Table 5 presents results obtained from 10 geographically diverse VPSes across Asia, Europe, and the Americas, each capturing three snapshots to mitigate location bias. As shown in the table, changing the VPS vantage points does not introduce a significant impact on the probing results, and the resulting topology scale remains comparable to that reported in Table 4.

Table 4: Snapshots of the Internet topology

Snapshot	Nodes	Links
1	2,159,102	7,163,165
2	2,608,245	8,692,028
3	1,892,796	6,518,245
4	2,448,310	7,322,468
5	2,191,159	7,250,477
6	2,561,608	7,808,499
7	2,191,159	7,090,131
8	1,930,050	6,811,102
9	2,262,536	6,902,365
10	1,816,264	7,001,376
11	2,603,425	7,674,352
12	1,992,157	6,563,335
13	2,372,374	7,340,365

Table 5: Snapshots of the Internet topology from 10 geographically diverse VPSes across Asia, Europe and the Americas

Snapshot	Nodes	Links
1	2,582,967	7,691,130
2	2,816,538	7,912,371
3	2,569,127	7,324,151

F Discussion

Diamonds caused by anycast. Diamond-like structures may also be observed in the presence of IP anycast, since the same destination prefix is advertised from multiple locations in anycast routing. This can cause that multiple parallel paths converge to a single destination IP, even though the destination IP is geographically or topologically distinct. To this end, we use prefixes appeared in the public anycast dataset [36] to exclude diamonds caused by anycast.

Ethics. Similar to previous research that uses active measurements, this work involves probing the Internet and takes precautions to minimize potential risks associated with such probing. To avoid disrupting the Internet, particularly to prevent overloading remote networks, we limit each VPS to a probing rate of 200,000 packets per second and randomize the probing sequence. Additionally, we set a PTR record for the IP addresses of each VPS, allowing network operators to contact us if they wish to opt out of the scanning. No other aspects of this work raise ethical issues.

BGP/route changes. During each snapshot, BayWatch uses historical observations to predict path structures, including potential changes due to BGP updates or intra-domain routing shifts. When a routing change remaps entire segments of a path, BayWatch selectively reverts probing for those hops to its bootstrap mode. For minor changes such as the addition or removal of a small number of interfaces or links, BayWatch continues in predictive mode without full reset. This adaptive fallback ensures robustness to routing dynamics while avoiding unnecessary probing overhead.

IPv6 support. BayWatch is equally applicable to IPv6 by simply targeting IPv6 prefixes, as the inference process itself is protocol-agnostic. Moreover, given the vast IPv6 address space, scaling Internet-wide probing in IPv6 remains an open challenge and merits a dedicated study.

G Gini Coefficient as a Measure of Load-Balancing Uniformity

Impact of Gini Coefficient on Discovery. To quantify how load-balancing skew affects probing effectiveness, we analyze real probing results from multipath structures with ten successors. In our dataset, we select nodes exhibiting ten-way load balancing and measure how many successors are covered under a fixed probing budget. The probing budget is set to the first 57 probes, which corresponds to the number of probes required by traditional MDA to discover all successors under the uniform load-balancing assumption.

Using this fixed probe budget, we evaluate discovery performance under different levels of skew. When the Gini coefficient is 0.2, 0.3, 0.4, and 0.5, the average numbers of successors discovered by the first 57 probes are 9.63, 8.06, 7.21, and 5.33, respectively. These results show that discovery effectiveness degrades significantly as load balancing becomes more uneven.

Interpreting Different Gini Levels. To provide intuition on what different Gini values represent, we extract representative probability distributions from real probing observations (ten successors, sorted in descending order). These distributions illustrate how load balancing progressively becomes more skewed as the Gini coefficient increases.

When the Gini coefficient is low (e.g., G=0.2), traffic is relatively evenly distributed:

$$(0.14,0.13,0.12,0.11,0.10,0.10,0.09,0.08,0.07,0.06). \quad (16)$$

In this regime, most successors receive comparable probability mass, allowing probes to quickly cover nearly all paths.

Under moderate skew ($G\approx$0.3), the distribution begins to exhibit a heavier tail:

$$(0.24,0.17,0.12,0.10,0.09,0.08,0.07,0.06,0.04,0.03). \quad (17)$$

A small subset of dominant successors absorbs a larger fraction of probes, while low-probability successors become slower to discover.

With stronger skew ($G\approx$0.4), probability mass concentrates further:

$$(0.32,0.20,0.13,0.10,0.08,0.06,0.04,0.03,0.02,0.02). \quad (18)$$

Several successors now receive very small probabilities, making them difficult to observe within a limited probing budget.

Under heavy skew ($G\approx$0.5), load balancing becomes highly uneven:

$$(0.45,0.23,0.12,0.07,0.05,0.03,0.02,0.015,0.01,0.005). \quad (19)$$

A few dominant successors capture the majority of probes, while low-probability paths are rarely sampled, explaining

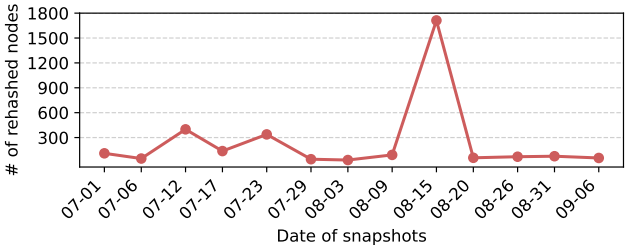


Figure 12: Number of rehashed nodes belonging to Zayo.

the sharp decline in discovery performance observed earlier.

These real-world examples demonstrate how increasing Gini directly translates into stronger load imbalance, reducing the likelihood of discovering low-frequency successors under a fixed probing budget.

H Case Study: Detecting a Real-World Network Anomaly

During continuous Internet monitoring, we pay particular attention to nodes where the LBR distribution prediction accuracy remains consistently below average. By tracking behavioral changes that deviate from the prediction, BayWatch can uncover anomalous network events.

A notable incident occurred on August 15, 2025, when we detected a large-scale rehashing event within a subset of Zayo's network. Specifically, nodes at the same hop began returning different next-hop addresses for identical flow identifiers, which is a clear deviation from their usual stable forwarding behavior. As shown in Figure 12, the number of rehashing nodes in Zayo's network exhibits a sharp spike in the snapshot of August 15, 2025, far exceeding its normal baseline.

To validate this anomaly, we cross-referenced the timestamp with external reports. According to Catchpoint [12], on the same day, ServiceNow experienced a two-hour outage due to unstable connectivity with its upstream provider, Zayo (AS6461). The incident led to degraded response times and intermittent global availability. This alignment between our internal signals (a sudden surge in rehashing events) and external evidence (a widely reported outage) illustrates how BayWatch can function as an effective early-warning system. That is, BayWatch can link topology anomalies to service disruptions, demonstrating the utility of BayWatch beyond topology discovery.