A Data-Driven Framework for Shared Bottleneck Detection across SD-WAN Overlays

Alessio Botta, Roberto Canonico, Annalisa Navarro,
Lorenzo Pitacoro, Giovanni Stanco, Giorgio Ventre, Stefania Zinno
DIETI Department, *University of Napoli Federico II*, Naples, Italy
{alessio.botta, roberto.canonico, annalisa.navarro, giovanni.stanco, giorgio.ventre, stefania.zinno}@unina.it,
l.pitacoro@studenti.unina.it

***Abstract*—Software-Defined Wide Area Network (SD-WAN) solutions rely on multiple overlays to enhance resilience, bandwidth utilization, and routing flexibility. However, because overlays are decoupled from the underlying physical network (underlay), shared underlay links among distinct overlays can remain hidden from the SD-WAN controller, possibly leading to congestion upon their concurrent utilization. Correlation of one-way delay (OWD) measurements across overlays has been proposed to detect shared bottlenecks without explicit underlay knowledge, yet the accuracy of this approach depends critically on two overlooked parameters: the volume of probing traffic and the correlation threshold. If not properly tuned, these parameters can lead to missed detections, false positives, or unnecessary network overhead. In this work, we address these challenges by introducing a traffic load calibration framework combined with a data-driven correlation threshold selection method. We evaluate our approach in an emulated real-world WAN scenario with multiple overlays, showing that it accurately identifies shared underlay resources while minimizing probing-induced congestion. Our results demonstrate that proper calibration significantly improves bottleneck detection effectiveness in SD-WAN.**

***Index Terms*—SD-WAN, Overlay Networks, Bottleneck Detection, Network monitoring**

I. Introduction

The increased deployment of Software Defined Wide Area Network (SD-WAN) in the last few years has significantly improved connectivity for enterprises and cloud providers worldwide. SD-WANs simplify network management by relying on multiple overlay connections as logical links between Edge Routers (ERs), all traversing the same physical infrastructure, referred to as the *underlay*. While overlays are managed by the SD-WAN controller, the underlay comprises the heterogeneous network over which traffic is physically routed [1].

The use of overlays offers significant benefits, including resilient backup paths, bandwidth aggregation, and flexible routing. However, because overlay and underlay are decoupled, the controller lacks both visibility and control over the underlay [2]. This becomes problematic when overlays selected by the controller share one or more underlay links: if a shared link experiences congestion, end-to-end performance can degrade significantly compared to using fully disjoint overlays. In such cases, detecting hidden shared network resources becomes essential for maintaining performance and avoiding bottlenecks.

A. Related Work

To improve fault tolerance and implement load balancing mechanisms, traditional routing algorithms have been enhanced to support routing over multiple links (multipath routing, i.e.), while guaranteeing link-disjoint path selection [3]. In SD-WAN scenarios, algorithms struggle to reach this goal as they depend primarily on knowledge of the network topology, which in SD-WAN is hidden due to the decoupling of overlay and underlay. Regardless of this limitation, SD-WANs are widely acknowledged for their efficient load balancing across multiple overlays [4], and various solutions have been developed to exploit this feature [5], [6]. The drawback of such approaches is that they react only after congestion has occurred and do not prevent bottlenecks proactively. Proactive overlay routing could efficiently avoid congestion when information about the underlying network topology is available. In this setting, traceroute-based probing or network tomography can be applied to map the network topology. Reconstructing the path from a source to a destination using **traceroute** with its packet probes proves to be the simplest approach. *Reverse Traceroute* is a more advanced approach that enables the direct reconstruction of the reverse path [7], applicable to networks of any scale, such as the Internet [8]. However, underlying routers do not always provide support for both traceroute and reverse traceroute. On the other hand, **network tomography/inference** techniques do not require intermediate routers' cooperation [9] and infer the internal network state by leveraging active monitoring. Network tomography can have two different goals [10]. Network performance tomography tries to determine link metrics, while network topology tomography aims to understand network topology, leveraging end-to-end measurements and without any previous knowledge. Active monitoring approaches do not rely on the underlying routers and have already proven suitable to monitor link failures [11] or latency [12] in SD-WANs. In particular, the correlation of One Way Delay (OWD) time series of flows crossing overlays demonstrates a powerful means to identify potential shared resources [13].

B. Contribution

However, prior works have not examined two critical aspects that influence the accuracy of correlation-based bottleneck

Stage 1: Traffic Load Calibration

Goal: Identify the proper load λ^* for measurements.

Procedure:

1) Initialize load $\lambda \leftarrow \lambda_0$.
2) Iteratively increase λ by step $\Delta\lambda$.
3) For each λ, perform N probing campaigns to measure one-way delays (OWDs) and compute pairwise correlation of OWDs.
4) Calculate correlation variability $\bar{\sigma}_{\text{corr}}(\lambda)$ across campaigns.
5) Stop increasing λ when $\bar{\sigma}_{\text{corr}}(\lambda)$ starts to rise.
6) Set λ^* as the last stable load and save correlation matrix C_{λ^*}.

C_{λ^*}, λ^*

Stage 2: Threshold Selection

Goal: Identify the proper threshold θ for measurements.

Procedure:

1) Extract the set of unique correlations $\mathcal{C}$ from C_{λ^*}.
2) Apply k-means ($k = 2$) on $\mathcal{C}$ to identify two clusters.
3) Select θ as the value maximally distant from cluster centroids c_1 and c_2.

Outputs: calibrated load λ^* and threshold θ.

Fig. 1: The framework first finds the suitable load λ^* through correlation variability, then derives a robust threshold θ through clustering; these parameters guide classification.

detection: (*i*) the level of probing traffic needed to guarantee meaningful correlation values, and (*ii*) the correlation threshold that best distinguishes whether overlays are joint or not. With this work, we demonstrate that without proper tuning of these parameters, detection can either miss existing shared bottlenecks or detect shared bottlenecks when they are not present, in addition to leading the network to excessive congestion. Moreover, we address these gaps by proposing a *traffic load calibration framework* combined with a *data-driven correlation threshold selection* method. Crucially, the framework is intended to tune these parameters directly within the operational network.

We test our framework in an emulated real-world WAN with multiple overlays and demonstrate that our approach identifies the optimal probing load and corresponding correlation threshold for reliable detection of shared underlay resources in SD-WAN environments, without excessive network overhead.

II. Correlation-based Framework

Probing is employed to explore the relationship between network overlays by generating different flows, each traversing a separate overlay. For instance, let f_u and f_v denote two such flows, with $l_u(t)$ and $l_v(t)$ representing their respective OWD time series. To understand whether these overlays share physical resources, we compute the Pearson correlation coefficient $c(l_u, l_v)$ between the two latency sequences and take its absolute value:

$$|c(l_u, l_v)| \geq \theta \tag{1}$$

If the correlation exceeds the threshold θ, the overlays are classified as *joint*, meaning that the flows likely share at least one common link. Otherwise, they are considered *disjoint*, indicating independent paths. The intuition is that flows traversing common network segments are subject to the same delay variations, leading to highly correlated latency patterns. In contrast, flows on distinct paths experience uncorrelated or weakly correlated delays. In the following subsections, we detail the two-stage framework for Traffic Load Calibration and Threshold Selection, as also described in Fig. 1.

A. Traffic Load Calibration

A traffic load calibration algorithm is necessary since, if the load is too low, the delay patterns may not be sufficiently pronounced to reveal shared bottlenecks. On the other hand, if the load is too high, congestion effects may dominate, introducing correlations that do not reflect actual path overlap.

At different load levels, the algorithm conducts probing campaigns, gradually increasing the traffic load λ. For each campaign i, the correlation matrix $C^{(i)}$ is derived from the correlation of the latency time series of every pair of flows.

From these multiple campaigns at load λ, the mean correlation

$$\mu_{\text{corr}}^{(\lambda)}(f_u, f_v) = \frac{1}{N}\sum_{i=1}^{N} C_{uv}^{(i)}$$

and standard deviation

$$\sigma_{\text{corr}}^{(\lambda)}(f_u, f_v) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(C_{uv}^{(i)} - \mu_{\text{corr}}^{(\lambda)}(f_u, f_v)\right)^2}$$

are derived for each flow pair (f_u, f_v) across the N campaigns.

For every load level λ, the mean standard deviation across all correlation values is calculated:

$$\bar{\sigma}_{\text{corr}}(\lambda) = \frac{1}{M}\sum_{(u,v)} \sigma_{\text{corr}}^{(\lambda)}(f_u, f_v)$$

where M is the total number of flow pairs.

The iterative load increase continues until $\bar{\sigma}_{\text{corr}}(\lambda)$ starts to increase, pointing to congestion effects that negatively impact measurement stability. The calibrated traffic load λ^* is then selected as the highest load level before this increase, representing the last stable measurement configuration.

B. Threshold selection

Rather than being fixed *a priori*, the threshold θ is obtained from the empirical distribution of correlations over all observed flows.

Given the collected flows in a probing campaign with the selected load $\lambda*$, and given $C \in \mathbb{R}^{N\times N}$ the correlation matrix of their latency time series, where:

$$C_{uv} = |c\,(l_u, l_v)\,|, \quad u \neq v, \qquad C_{uu} = 0 \tag{2}$$

The unique correlation values make a set $\mathcal{C}$ defined as:

$$\mathcal{C} = \left\{ C_{uv} \;\middle|\; 1 \leq u < v \leq N \right\} \tag{3}$$

Fig. 2: Experimental setup based on the Abilene backbone topology [14]. Red markers indicate SD-WAN ERs instantiated as Open vSwitches. ERs are connected to Mininet hosts generating probing flows to measure overlay one-way delays.

k-means clustering, with $k = 2$, is applied to $\mathcal{C}$, and c_1 and c_2 are the resulting centroids. The threshold θ is then selected as the value that maximizes its distance from both centroids. The decision criterion follows a data-driven approach and is tailored according to the correlation structure present in the network to generalize to different operational scenarios.

III. Evaluation

1) Experimental Setup: The equipment on which evaluations were performed consisted of an Ubuntu 20.04-based virtual machine, provisioned with a 3-core processor and 8 GB of RAM. The Mininet emulator was employed for reproducing the WAN scenario. Traffic generation and OWD measurements were carried out using *Distributed Internet Traffic Generator* (`D-ITG`) [15].

2) Network Topology: For our experiments, we adopted the Abilene backbone topology [16]. Propagation delays between edge nodes were derived from their geographical coordinates, varying between 0.22 ms (minimum) and 1.7 ms (maximum), with an average of 0.90 ms. Bandwidth is also set to 2 Mbps across couples of edges. Each red marker depicted in Fig. 2 represents an ER at an SD-WAN endpoint. In our setup, ERs were instantiated as Open vSwitch (OvS) nodes. ERs are connected to Mininet hosts acting as probing endpoints, which send measurement flows to estimate the OWDs of the overlays.

3) Experiment Parameters: Within the Abilene topology, one host was configured as the sending probe and nine as receiving probes, resulting in traffic transmission of nine flows over nine distinct overlays. The probing flows consisted of UDP packets having a constant size of 512 bytes. Results are reported for five per-flow traffic rates λ: 50, 100, 125, 150, and 200 pkt/s. Mean latency values for the time series were calculated over $I = 1$ s intervals by averaging the delays of all packets received in that window. Each measurement session lasted $T' = 60$ s, and for every load rate λ, $N = 3$ independent repetitions were conducted.

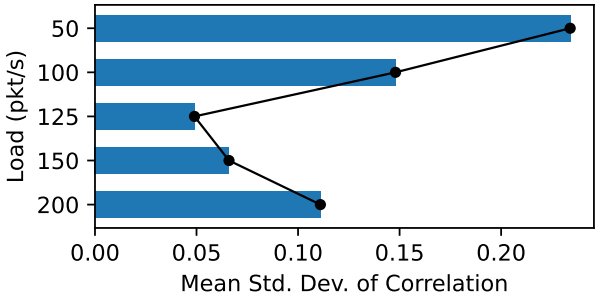


Fig. 3: Correlation variability increasing load

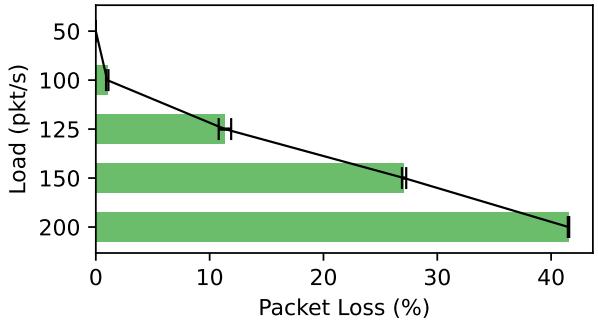
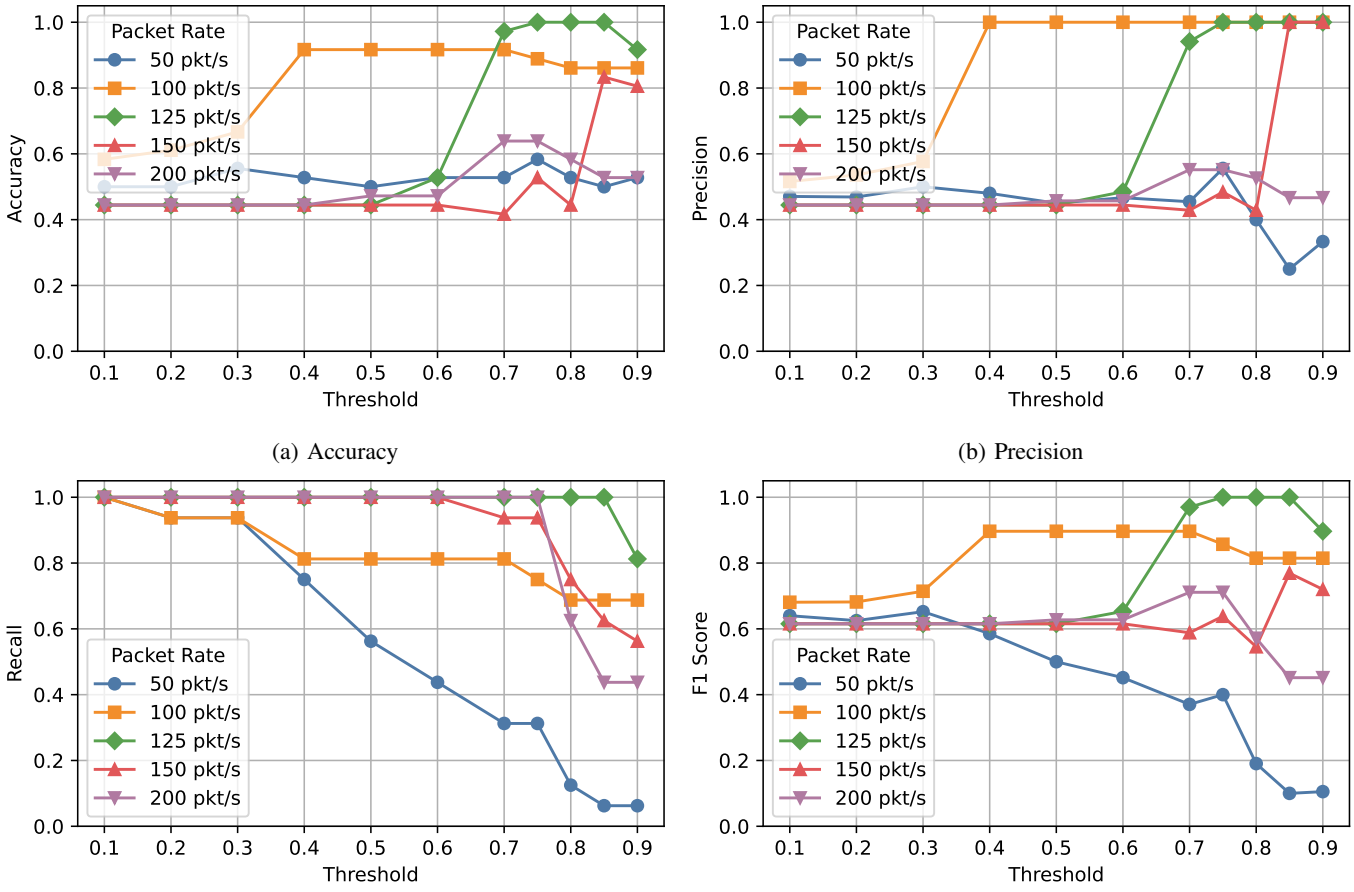


Fig. 4: Packet loss increases with probing load

4) Evaluation Metrics: Overlay paths were classified as *joint* (i.e., using at least one physical link in common) or *disjoint* (i.e., having no links in common) according to the correlation values between the OWD time series of concurrent probing flows. Classification errors have distinct consequences: a False Positive (FP), where a disjoint pair is incorrectly labeled as joint, may lead to underutilization of network resources, whereas a False Negative (FN), where a joint pair is mistakenly labeled as disjoint, can cause congestion when both overlays are used simultaneously. To assess classification performance, we consider precision, recall, F1-score, and accuracy, capturing both the correctness and the trade-off between congestion avoidance and bandwidth utilization.

5) Results: Following the traffic load calibration framework, we compute the mean and standard deviation of correlation values for each flow pair across multiple experiments and derive the variability of the correlation for all tested load rates (Fig. 3). A high variability means that correlations fluctuate significantly between repetitions at the same load, making that load unsuitable for reliable inference. Conversely, low variability indicates stable correlation patterns, enabling consistent structural analysis. Results show high variability at low loads ($\lambda = 50, 100$ pkt/s), where limited interference prevents stable correlations from emerging. Variability decreases and reaches a minimum at 125 pkt/s, where resource contention stabilizes delay relationships. Beyond this point, congestion increases variability again, reducing the discriminative power of correlation values. The selected operating point, $\lambda^* = 125$ pkt/s, is selected according to the framework, since



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-score

Fig. 5: Classification performance vs. correlation threshold (θ) and packet rate (λ). The best results occur at $\lambda = 125$ pkt/s and $\theta \approx 0.8$, while $\lambda = 100$ pkt/s shows more stable performance across thresholds. Higher loads require larger θ values but eventually degrade performance due to congestion.

the correlation variability starts increasing after this load rate. As shown in Fig. 4, packet loss increases with the load level. In particular, it is negligible at low loads, moderate at 125 pkt/s ($\approx 11.3\%$), and excessive at high loads. The selected load avoids significant network congestion, which tends to occur at higher load levels.

6) Overlay Classification: We assess the performance of our classification problem under varying values of the correlation threshold θ and traffic rate λ. We depict the algorithm performance in the subfigures of Fig. 5, showing accuracy, precision, recall, and F1-score at different packet rates.

An analysis on **accuracy** (Fig. 5a) indicates that performance is poor when the traffic load is low ($\lambda = 50$ pkt/s), due to insufficient traffic intensity, which limits the emergence of meaningful correlations. Higher values of λ result in higher accuracy: in particular, the best performance is reached for $\lambda = 125$ pkt/s, where the selected load rate allows reliable detection of joint overlays. In particular, for these two load levels, the accuracy improves increasing θ, and the best performance is reached in the first case for $0.4 < \theta < 0.7$ and in the second case for $0.7 < \theta < 0.85$. Accuracy declines again at $\lambda = 150$ pkt/s and $\lambda = 200$ pkt/s because of spurious correlations caused by congestion.

Precision (Fig. 5b) remains low across various threshold values at low load ($\lambda = 50$ pkt/s). As load increases to 100 pkt/s and 125 pkt/s, precision improves with increasing θ, since the number of disjoint overlays misclassified as joint (i.e., the FPs) decreases and reaches the maximum in the first case for $0.4 < \theta < 0.7$ and in the second case for $0.7 < \theta < 0.85$. At the highest loads, precision is again low for all thresholds due to network congestion.

Examining **recall** (Fig. 5c), low thresholds yield the highest recall at low traffic loads, as most flows are classified as joint. As θ increases, recall declines across all load levels; however, the decrease is markedly steeper for low loads, while higher loads are able to sustain comparatively higher recall values even at larger thresholds. Higher loads ($\geq 125\ pkt/s$) manage to get the highest values for recall.

The **F1-score** (Fig. 5d), which balances precision and recall, is low at low load and decreases as the threshold increases.

It improves at intermediate loads, achieving the best value at $\lambda = 125$ pkt/s for θ around 0.8, and then decreases again for high loads. We notice that while $\lambda = 125$ pkt/s yields the best results in both Accuracy and F1-score, $\lambda = 100$ pkt/s is the load value that is more robust to the choice of the threshold, getting the best performance for a higher threshold range.

In summary, increasing traffic load necessitates a higher correlation threshold θ to accurately discriminate joint overlays, as stronger correlations emerge under heavier traffic. However, excessive load causes congestion and degrades performance, while a load level that is too low is also not useful for classification, underscoring the importance of choosing appropriate load levels and thresholds.

TABLE I: Comparison between framework-selected (Ours) and optimal (Opt.) thresholds with corresponding performance metrics for different traffic loads.

Load (pkt/s)	θ	θ **value**	**Accuracy**	**Precision**	**Recall**	**F1**
50	Ours	0.54	0.50	0.45	0.56	0.50
	Opt.	0.3	0.56	0.50	0.93	0.65
100	Ours	0.57	0.92	1.0	0.81	0.90
	Opt.	$[0.40, 0.70]$	0.92	1.0	0.81	0.90
125	Ours	0.80	1.00	1.00	1.00	1.00
	Opt.	$[0.75, 0.85]$	1.00	1.00	1.00	1.00
150	Ours	0.82	0.44	0.43	0.75	0.55
	Opt.	0.85	0.83	1.00	0.63	0.77
200	Ours	0.86	0.53	0.47	0.44	0.45
	Opt.	$[0.70, 0.75]$	0.64	0.55	1.00	0.71

Table I reports, for each load, the performance metrics corresponding to the optimal threshold (or threshold range), i.e., the one with the highest F1-score based on the ground truth, and the one chosen by our data-driven threshold selection method with k-means. A traffic load of 125 pkt/s allows for achieving the best overall performance, with perfect classification results and the four Precision, Recall, Accuracy, and F1-score metrics all equal to 1. For this load level, the framework-based threshold falls within the optimal one, demonstrating the effectiveness of our framework. Also, for the other load levels besides the first and the last, the ones with low detection performance, the chosen threshold falls within the optimal range or differs minimally from the optimal value.

IV. Conclusion

In this work, we addressed the challenge of detecting shared bottlenecks in SD-WAN overlays, where the decoupling of overlay and underlay hides critical topology information from the controller. We leveraged the correlation of one-way delay (OWD) time series between overlay paths to identify potential shared underlay resources and proposed a framework that combines traffic load calibration with data-driven correlation threshold selection. This enables accurate detection without introducing excessive probing overhead. Experimental results demonstrate that both parameters are key to achieving high detection accuracy, and that our framework enables systematic tuning without prior knowledge of the physical topology. By identifying the optimal probing load and correlation threshold, the proposed approach improves correlation-based shared bottleneck detection. Future work will evaluate the benefits of an underlay-aware SD-WAN controller that, when possible, selects disjoint overlays to improve QoS.

Acknowledgment

This work was partially supported by Cisco Systems through the Sponsored Research Agreement "Research Project for Industry 4.0", and by the European Union – Next Generation EU, Mission 4, Component 1, through the ADAPTO project, part of the RESTART program, CUP E63C2 2002040007, CP PE0000001.

References

[1] S. Troia, L. Borgianni, G. Sguotti, S. Giordano, and G. A. Maier, "A Comprehensive Survey on Software-Defined Wide Area Network," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, 2025, accepted July 26, 2025.

[2] S. Troia, J.-P. Asdikian, G. Sguotti, E. Gregorini, M. Li, and G. Maier, "In-Band Network Telemetry for software-defined wide area networks," *Computer Networks*, p. 111567, 2025.

[3] D. Lopez-Pajares, E. Rojas, J. A. Carral, I. Martinez-Yelmo, and J. Alvarez-Horcajo, "The disjoint multipath challenge: Multiple disjoint paths guaranteeing scalability," *IEEE Access*, vol. 9, pp. 74 422–74 436, 2021.

[4] S. Rajagopalan, "An overview of SD-WAN load balancing for WAN connections," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2020, pp. 1–4.

[5] Y. Zhang, J. Tourrilhes, Z.-L. Zhang, and P. Sharma, "Improving SD-WAN resilience: From vertical handoff to WAN-aware MPTCP," *IEEE transactions on network and service management*, vol. 18, no. 1, pp. 347–361, 2021.

[6] S. Lee, K.-Y. Chan, and T.-Y. Chen, "Design and implementation of an sd-wan vpn system to support multipath and multi-wan-hop routing in the public internet," *Authorea Preprints*, 2023.

[7] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. E. Anderson, and A. Krishnamurthy, "Reverse traceroute." in *NSDI*, vol. 10, 2010, pp. 219–234.

[8] K. Vermeulen, E. Gurmericliler, I. Cunha, D. Choffnes, and E. Katz-Bassett, "Internet scale reverse traceroute," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 694–715.

[9] A. Coates, A. Hero III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal processing magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[10] T. He, L. Ma, A. Swami, and D. Towsley, *Network tomography: identifiability, measurement design, and network state inference*. Cambridge University Press, 2021.

[11] A. Botta, R. Canonico, A. Navarro, G. Stanco, and G. Ventre, "Towards a Highly-Available SD-WAN: Rapid Failover based on BFD Protocol," in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2023, pp. 153–158.

[12] ——, "Adaptive overlay selection at the SD-WAN edges: A reinforcement learning approach with networked agents," *Computer Networks*, vol. 243, p. 110310, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624001427

[13] D. Z. Tootaghaj, F. Ahmed, P. Sharma, and M. Yannakakis, "Homa: An efficient topology and route management approach in SD-WAN overlays," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2351–2360.

[14] M. Bloem, T. Alpcan, S. Schmidt, and T. Basar, "Malware filtering for network security using weighted optimality measures," in *2007 IEEE International Conference on Control Applications*. IEEE, 2007, pp. 295–300.

[15] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[16] M. Großmann and S. Schuberth, "Auto-Mininet: Assessing the Internet topology zoo in a software-defined network emulator," *Messung, Mellierung un Bewertung von Rechensystemen (MMBnet)*, vol. 7, pp. 1–10, 2013.