SpeedCam: Towards Efficient Flow Monitoring for Multipath Communication

Kilian Gärtner
Otto-von-Guericke-University Magdeburg
herr.kilian.gaertner@gmail.com

Jonghoon Kwon
ETH Zurich
jong.kwon@inf.ethz.ch

David Hausheer
Otto-von-Guericke-University Magdeburg
hausheer@ovgu.de

***Abstract*—Network monitoring is essential for traffic engineering, maintenance, and troubleshooting purposes and thus forms an integral part of network management. However, observing each and every packet may not be feasible or at least be very costly. It is therefore crucial for network operators to ensure a scalable and efficient monitoring. With the emergence of multipath communication as facilitated by new network architectures like SCION, monitoring becomes an even more challenging task. In a multipath network, operators need to be able to monitor their customers' traffic flows across different network paths to ensure optimal network utilization, fault tolerance and fairness. Traditional single-path flow monitoring such as sampling-based mechanisms fall short, since packets may be spread across a potentially large number of different paths. To address this challenge, this paper proposes SPEEDCAM, a new approach that aims to achieve scalable and efficient flow monitoring in multipath networks. Our approach which is based on probabilistic probe selection significantly reduces the number of required monitoring probes, while enabling an effective flow information gathering. With an implementation of SPEEDCAM in the SCIONLab network, we demonstrate more than 89 % of monitoring accuracy with a small fraction of network routers covering only 50 % of network traffic in the multipath network.**

***Index Terms*—Network monitoring, multipath communication, SCION**

I. INTRODUCTION

New networking paradigms have caught the attention of the networking industry, specifically multipath communication [1, 2]. Instead of using a single path for an end-to-end communication, multipath communication allows end-hosts to transmit packets via multiple paths simultaneously. By scattering packets through multiple paths, network operators are able to distribute the load over their network, achieving better utilization of the overall network capacity, and thus saving network provisioning costs. Furthermore, multipath communication allows fast failover in case of congestion, and thus improves network reliability. Driven by this, considerable ongoing efforts from both academia and industry are being made to realize these benefits [3, 4].

Despite these promising benefits, multipath communication also introduces new problems faced by network operators [5]. For example, a malicious user may abuse multipath communication to achieve better throughput or to disrupt the network by flooding packets through all possible paths. In such cases, extensive network monitoring and analysis are required to accurately detect the misuses.

978-3-903176-32-4

Today, network operators collect flow-level measurements for traffic engineering [6], QoS [7], and anomaly detection [8], but modern network routers hardly support per-packet recording. Driven by this, various sampling techniques have been suggested. Routers may collect packets or flows based on a sampling probability and aggregate them to conjecture the flow statistics [9, 10]. Although sampling-based flow measurement demonstrates viability of passive monitoring, it shows limitations with regards to monitoring multipath communication since correlating all the flows collected from densely deployed probes would introduce scalability issues.

To this end, we introduce SPEEDCAM, a scalable network monitoring approach for multipath communication environments. SPEEDCAM enables efficient multipath-flow measurements based on a probabilistic approach. More precisely, an inspector (e.g., network operator) selects a subset of routers called probes over which the inspectee's multipath-flows are observed with a high probability, and conducts an inspection at random times. The selection is performed based on the probability modeling; candidate scores for each node are measured based on the network topology and traffic statistics, reflecting the probability of being probes for effectively measuring the majority of multipath flows. The model keeps changing the list of probes, introducing unpredictability of the inspection.

We have implemented SPEEDCAM, which performs network exploration, probe selection, traffic monitoring and classification. In the evaluation, which has been conducted over the multipath-enabled SCIONLab network [11], we demonstrate that SPEEDCAM achieves more than 89 % of monitoring accuracy using only 33 % of routers with 50 % of network coverage. The overall evaluation introduces negligible performance overhead—the average resource consumption for CPU and memory is less than 0.5 and 2%, respectively.

In summary, this paper makes the following contributions:

- To the best of our knowledge, this is the first work that addresses network monitoring for multipath architectures.
- We present SPEEDCAM, a new approach enabling efficient and effective flow monitoring in multipath communication environments.
- We demonstrate the viability of SPEEDCAM through a proof-of-concept implementation and its evaluation in a real-world network testbed.

II. Related Work

Various network monitoring techniques have been developed, following the steps of measurement, aggregation and analysis. Such network monitoring can be classified into active and passive monitoring [12].

In active monitoring, the network operator generates an explicit control packet to check the status of data plane. The control packet is forwarded to the specific nodes performing the monitoring to execute the command and report the collected result to the operator. Active monitoring has attracted many researchers because it is more applicable and scalable than passive monitoring. In general, the goal of active monitoring is to cover an entire network using as few monitoring nodes as possible, achieving cost-efficiency [13]. The location optimization problem was found to be NP-hard problem; to overcome this greedy approximation algorithms were proposed [14].

Passive monitoring, unlike active monitoring, does not introduce control overhead. Monitoring devices continuously collect and store network traffic, and report it to network operators. Because of the rapid increase in network traffic, it is impossible to collect all the traffic, so sampling methods based on probabilistic models caught attention. Accordingly, approaches such as packet sampling [15] or flow sampling [9] have been actively investigated. The location optimization problem also occurs here [14].

In general, network monitoring follows the steps of measurement, aggregation, and analysis. First, network information collection basically considers how, where, and when to collect measurements [16]. The collected raw data is mapped into traffic patterns, network status, networking policies, etc., and then analyzed according to the purpose of monitoring. The collection is responsible for delivering the collected data to the analysis station. Methods to effectively deliver large amounts of data, such as SNMP [17] and Syslog [18], have been studied. Finally, in the analysis stage, traffic is classified using packet payload or flow information, and statistics or network status are analyzed. The results are then used for traffic engineering, anomaly detection, and troubleshooting.

Unfortunately, the traditional network monitoring approaches cannot be directly applied to multipath networks, which allow a flow to utilize various paths at the same time [19, 3, 4]. Traditional network monitoring is built on the assumption that a flow travels through a single path determined by the ISP. In contrast, multipath communication is often realized with the notion of path-aware networking [20, 21]. That is, ISPs provide end hosts a set of different paths leading to a destination, allowing them to choose and utilize the paths simultaneously for a single communication. The end-host driven path selection empowers applications with many desirable features such as path transparency, fine-grained path control, fast failover, and route optimization. For network administrators, however, the shift of decision making on path selection from network to end hosts causes challenges on network monitoring and management.

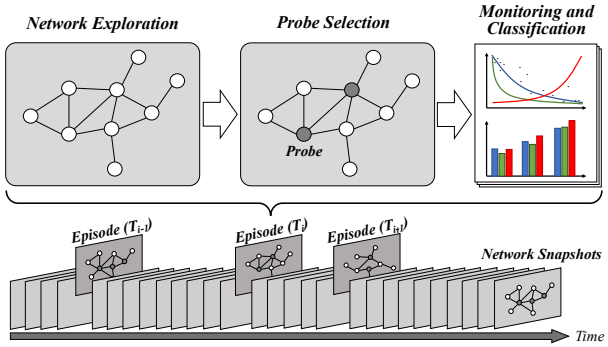

Fig. 1: High-level overview of SpeedCam.

III. System Overview

The aim of SpeedCam is to enable accurate network monitoring for multipath communication in a scalable and cost-efficient way. This is achieved by aggregating network snapshots based on probabilistic modeling and correlating them to detect abnormal activities in the network. We first outline the desired properties derived from previous works.

- **Scalability**: The approach should be lightweight in network monitoring, such that it enables scalable flow measurements regardless of the network topology, size and traffic volume.
- **Accuracy**: The probabilistic model should be able to discover the most effective vantage points that covers the majority of multipath communication.
- **Obscurity**: The probabilistic model must ensure that adversaries cannot infer the next inspection details based on the previous inspections or even the algorithm itself, such that it achieves unpredictability.
- **Practicality**: Probes should be able to conduct the monitoring without huge overheads that might affect the actual network performance.

A. General Concept

SpeedCam applies to networks which meet the following requirements. The network must consist of *nodes* and *links* which form an undirected graph. For SpeedCam, the graph does not necessarily need to be known or static. This is important because real networks tend to be dynamic with only partial knowledge about its structure. The inspector must have access to the network in a form that allows it to gain knowledge about the network structure and to monitor the traffic of the nodes.

When these requirements are met, the inspector can execute an inspection and aggregate the flow measurements for each inspectee. Note that in this work, the inspector is described as a centralized component, but it may be split into multiple inspectors, each responsible for a subnetwork. One run of the inspection is called an *episode* and consists of four phases as shown in Figure 1.

Exploration. This phase has the goal to create an undirected graph of the network to be monitored. The structure of the

network is required for the following phases. In the following it is assumed that the network graph can be created, e.g. by collected information from each node. Note that the exploration phase needs to be constantly repeated to react to changes in the network; the exploration phase precedes every episode by default, but it can be skipped if the network is static and fully explored.

Selection. Based on the results of the previous exploration phase, the selection phase uses the network information and returns a set of nodes, called *probes*, to monitor. The selection process has two goals to achieve: 1) it has to minimize the amount of used probes for scalability, while 2) maximizing the network coverage to achieve a high accuracy of the inspection. To achieve these goals, the selection uses the topology and information from past monitoring phases to increase the quality of a selection. The selection quality is defined as follows:

$$q(S, N) = \frac{\text{coverage}(S)}{\text{coverage}(N)} \tag{1}$$

where S represent a set of probes and N the set of all nodes.

Monitoring. The monitoring uses the selected probes from the previous phase. The inspector initiates the monitoring by triggering the probes to measure the flows forwarded through them. The probes log the flow measurements over a time span $\Delta T = t_2 - t_1$. At the end of the time span, the inspector aggregates the flow measurements.

Classification. In this phase, the aggregated flow measurements are intercorrelated to quantify the total amount of traffic that travels through different paths for the same source and destination pair.

B. Assumptions

We clarify the assumptions to help further understanding SPEEDCAM and the underlying environment.

- **Secure communication**: We assume that the communication between the centralized controller (e.g., inspector) and probes is secure. There are well-established mechanisms enabling secure-channel establishment that can be applied for the distributed environment.
- **Flow measurement**: We also assume that the probes can efficiently quantify the flows for the time span, and there is no degradation on packet forwarding performance.

IV. SPEEDCAM

To achieve scalable and accurate network monitoring for multipath communication, an essential component of SPEEDCAM is probe selection. To ensure a good network monitoring quality, we consider unpredictability in selecting probes covering the network as widely as possible.

Unpredictability. The inspector shall not be predictable for the inspectees. When a set of probes is predictable for the inspectees, they can avoid the inspection by choosing paths without probes. From this, we first present two strawman approaches, i.e., *obscurity centric* and *entropy centric*, and then provide our design that combines the two approaches.

The *obscurity-centric approach* counters the predictability by keeping the selection algorithm and its configuration secret from the inspectees. In this approach, the probe selection is focused on its effectiveness by applying heuristics, considering the number of probes, coverage of each probe, and the selection history. However, the unpredictability only lasts as long as the selection algorithm remains secret. The inspectees could use the inspection statistics to reverse-engineer the algorithm.

In *entropy-centric approach*, we can select random probes by using an entropy. Instead of selecting probes based on statistics, the algorithm randomly selects a set of probes, preventing predictability to inspectees. Nonetheless, there is a conflict between the randomness and effectiveness of probe selection; a fully random selection may result in a completely useless data collection on the multipath correlation.

Therefore, we combine the two approaches to achieve both effectiveness and randomness in probe selection. SPEEDCAM uses a heuristic approach called *candidate score*, denoted as $cs(n)$, to determine the probability of each node to be chosen as a probe. While not guaranteed to be chosen, the chance to be a probe is higher for a good candidate than for a bad candidate. Then, we randomly select k probes from a set of best candidates. This process will result in a high quality of the probe selection concerning the network coverage while at the same time not being fully deterministic.

A. Candidate Score

The following criteria can be calculated using the information collected during the *Exploration* phase.

Degree. [$deg(n)$]: The total number of incoming and outgoing links of a node. This criterion indicates the likelihood of a good transport hub and can be used as a central transport hub by other nodes. The higher the degree of a node, the higher the possibility to monitor a great range of traffic sources. Leaf nodes on the edge of the network use to have a low degree and are therefore uninteresting as probes, since they may only monitor traffic from/to these nodes.

Total capacity. [$cap(n)$]: The sum of the connection capacity of a node. The capacity is the physical or agreed connection limit between partners. This is similar to the degree, but it differs in one aspect: The degree is only an indicator about the possible connectivity, but not for the possible throughput of traffic. A node with a high capacity may be preferred over a node with a high degree and lower capacity to transfer data.

Average activity. [$\overline{act_{t_1,t_2}}(n)$]: The relation between potential and actual throughput of a node in a time interval between t_1 and t_2. The time span can be of any resolution, but should be consistent.

$$\overline{act_{t_1,t_2}}(n) = \frac{\text{traffic}_{\Delta T}(n)}{cap(n)} \tag{2}$$

Storing the timespan for an episode gives the possibility to create a profile for a node to be used by the selection process. The selection itself can be time based. A node with a high activity in the morning may receive a higher candidate score than a node with a high activity in the evening, which can increase the quality of the probe set.

Finally, the candidate score for each node is calculated as follows:

$$cs(n) = w_d \cdot deg(n) + w_c \cdot cap(n) + w_a \cdot \overline{act_{t_1,t_2}}(n) \quad (3)$$

where w means weight for each attribute ($w \geq 0$).

The average activity will be zero for the first episode, because the inspector does not have information about the traffic of such a node. Note that, we set the weights equally in our evaluation, but the weights can be configured by network operators to address various purposes of network monitoring.

Probability. The probability to be selected as a probe depends on the $cs(n)$. It is calculated for all nodes in the network. With that information, the probability can be calculated and assigned to the node with:

$$P_{sc}(n) = \frac{cs(n)}{\max\left(cs(N)\right)} \quad (4)$$

where N denotes all nodes in the network.

B. Probe Selection

Based on the list of candidates with the assigned selection probability (candidate score), the inspector selects k probes. The size of k should scale with the network size $|N|$, but as discussed earlier it should be as small as possible. The question in this phase is how many nodes does the inspector need to deploy to find a greedy user with a high accuracy? There are a few possible scales, that we discuss in the following:

Constant. [K]: A constant amount of nodes will be selected, ignoring the size of the network. The advantage is the predictability of the computation time suitable for static networks. However, it is not sufficient for a dynamic network growing over time.

Logarithmic. [$log(|N|)$]: A logarithmic amount of probes will scale nicely with any network because it needs very few nodes in large networks.

Linear. [$x \cdot |N|, 0 < x \ll 1$]: A linear amount of probes will need more resources than the logarithmic approach, but it will also raise the accuracy to sufficient levels. The factor x must be much less than 1, otherwise the inspector will utilize half of the network which contradicts the goal to use as few resources as possible. The evaluation will use $x = \{0.1, 0.2, 0.3\}$.

The selection phase ends with the set of k nodes, used for the next phases as probes.

V. Evaluation

We conducted our SpeedCam experiment for 48 hours on SCIONLab testbed [11], a novel Internet testbed that supports native multipath communication. The testbed includes 38 infrastructure nodes and hundreds of user-created nodes physically distributed across 4 different continents. Since SCIONLab is a live testbed where users continuously create and destroy their nodes for research purposes, the network topology keeps changing over time. Thus, it provides a dynamic network environment for testing purposes. Figure 2 shows a snapshot of the SCIONLab topology that consists of a global backbone, 10 core ASes, and more than 100 non-core ASes.

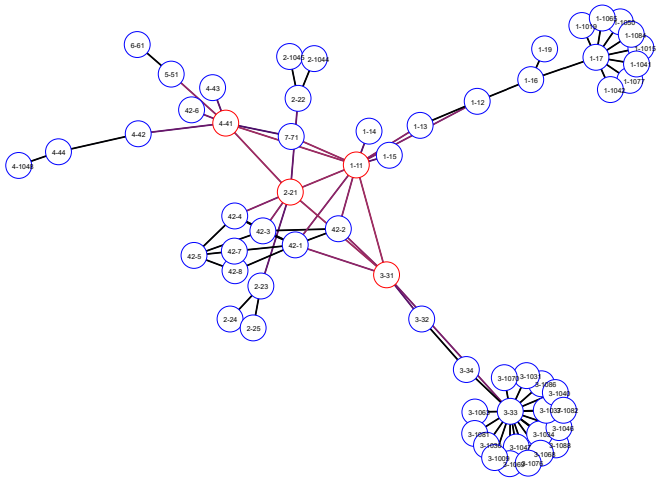

Fig. 2: A snapshot of SCIONLab. The red nodes are selected probes and the red lines indicate the links under monitoring.

The prototype (https://github.com/Meldanor/SCIONLab_SpeedCam) ran on a VM based on an Intel i7@3.75 GHz with one assigned core, 1GB of RAM and a 52GB SSD. This VM hosted also the Prometheus server in version 2.2.1. It provided a ground truth of the measured traffic inside the network. This instance fetches data from all nodes in an interval of 15 seconds. The difference to the inspector is that the Prometheus server fetches always all the metrics. This data is used to calculate the quality of the selection, as described in Equation 1. For further information about the SpeedCam implementation, please refer to [22].

A. Scalability Analysis

To achieve scalable monitoring, it is important to cover a wide range of the network with a small number of monitoring assets. We first investigate the monitoring precision, i.e., the proportion of the traffic volume monitored by the selected probes. Figure 3 shows the cumulative distribution function of the precision for the different probe-selection strategies, where N is the total number of nodes and k is the number of selected probes. Note that only 21 nodes out of the approximately 100 active nodes have shown a high enough capability to monitor the traffic passing through. Thus, in this experiment, we consider $|N| = 21$.

For comparison, we set *const* number of probes with $k = 1$ as an extreme case. The single probe results in the worst traffic coverage of 10 – 20 %, but it is a surprising result for only one probe. According to our investigation, SpeedCam keeps selecting a node in core networks; 3,505 times out of 11,575 inspection rounds. The selected probes are connected to other core networks, thus responsible for a high volume of traffic at inter-domain level.

As expected, the higher the number of probes the better the overall precision. $k = log(N)$ uses only two nodes resulting in an increase of 10 % precision. A similar result

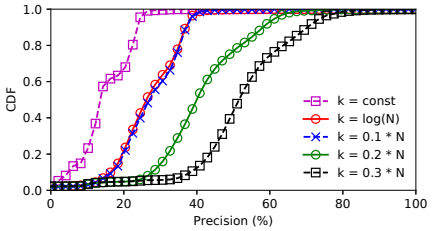

Fig. 3: CDF of the monitoring precision for the different selection strategies.

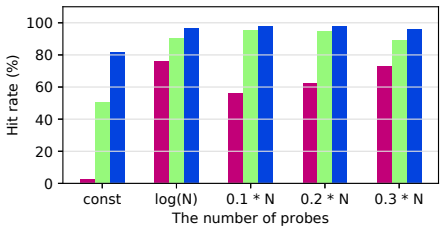

Fig. 4: Min., Avg., and Max. hit rates for the different selection strategies.

TABLE I: Average resource consumption (%) with snapshots every 5 seconds.

	CPU	Memory
$k = 1$	0.1	1.1
$k = log(N)$	0.2	1.1
$k = 0.1 * N$	0.2	1.3
$k = 0.2 * N$	0.3	1.3
$k = 0.3 * N$	0.4	1.2

is given by $k = 0.1 * N$, that selects the same number of probes for inspections. The other linear scales, $k = 0.2 * N$ and $0.3 * N$, achieve better precision by using more nodes, four and six probes, respectively. Interestingly, the linear configurations show sometimes 90 – 100 % of precision with only 4 – 6 probes. The reason for that phenomena is that the probes are well distributed across the core networks and well positioned without overlapping links.

B. Accuracy Analysis

The hit rate is an important criteria to evaluate the quality of each selection strategy. It indicates how accurately a target flow is measured by the inspection. Therefore, a high precision but a low hit rate may lead to a bad decision for the monitored network. Figure 4 illustrates [Min., Avg., Max] values of the hit rate for the different selection strategies. To measure the hit rate, we target a specific flow that occupies the maximum capacity in the network at each episode.

Overall, regardless of the number of probes, the hit rates show a similar result in average, approximately 89 – 95 %. Differences between the linear-scaling approaches appear mostly in the minimum hit rate, ranging in 56 – 72 % of hit rates. This is also expected since more probes have a higher chance to observe the traffic distributed across the network. Interestingly, the log-scale selection shows a relatively higher hit rate at its minimum despite the same number of probes compared to $k = 0.1 * N$. Nevertheless, this could happen since the testing environment is a live network; each episode has a different network snapshot.

With $k = 1$, in general, SPEEDCAM shows relatively low hit rates compared to other selection strategies. In the worst case, 2.59 % of hit rate is measured. That is, a node with a low $cs(n)$ is selected as the probe, so that it could not gather the target flow. However, it scores approximately 50 % of hit rate in average, which is also surprising considering that there is only one probe in the network.

From these results, we make the following observations: 1) in general, the number of probes has a relatively small effect on the hit rate. 2) the hit rate can be varying depending on the network dynamics. 3) the precision is not directly related with the hit rate. This suggests that the monitoring location is more important than the number of probes.

C. Practicality Analysis

The performance impact of SPEEDCAM on the system is minimal as seen in Table I. Without the outliers, no configuration uses more than 7 % of the CPU time; fewer than 0.5 % in average. The main impact on the CPU is the amount of probes as seen in the linear configuration. The more probes exist, the more candidate scores and the more monitoring results need to be calculated. The memory impact has a similar result. The resource consumption is negligible; No configuration needed more than 1.5 % of the memory.

VI. DISCUSSION AND FUTURE WORK

A. Inspection Interval

Each inspection is repeated in a certain interval, and there are different strategies possible.

With a *fixed* interval, the inspector repeats the inspection every t time units and gets a constant stream of information about the network traffics. The precision of the inspection might be different depending on t; a smaller interval results in a higher precision, but also increases/decreases the resource impact. Furthermore, it is possible for the inspectees to infer the inspection pattern.

A *randomly* chosen time interval prevents this problem. The inspector starts an inspection non-deterministically a few times over a given timespan, such as an hour or a day. This decreases the precision of the measurement, because the interval changes randomly. The inspector loses the possibility to improve its selection quality by using the time profiles. This strategy may waste the inspector resources, when the inspection is randomly chosen at a point of time when nothing intersting happens.

Another strategy is to utilize the knowledge of past episodes and the course of the traffic over a day. The inspector can check it when the traffic is usually high, for example in the evening. This *experiences-based* strategy has the same advantages and issues as the activity criteria from subsection IV-A. A longer history enables a more precise decision, but a too long history can hold outdated information. Also it is a purely experience based strategy vulnerable to a local optima.

B. Probe Selection using Control-plane Information

The probe selection has a significant impact on the quality of multipath flow monitoring. SPEEDCAM leverages network topology and traffic statistics that reflect the current state of a

given network. To improve the quality of probe selection, we can further utilize the control plane information.

In SDN, there exist the controllers in which requests for routing decision aggregate. For example, the Openflow switch communicates with the controller to program the flow table when a new flow arrives. SCION also has a control plane protocol called *path request*. When an end host establishes a communication channel, the host generates a control message to fetch end-to-end routing paths from a controller called the path server. The control plane information is essential because the information provides a brief overview of current user activities over the network. By feeding the selection algorithm with the control plane information, the quality of probe selection could be improved. Our future work includes investigating how the control plane information could improve the quality of probe selection.

VII. Conclusion

Multipath communication has brought a new challenge to network monitoring. For a scalable and cost-efficient monitoring of multipath flows, we have introduced a new framework that leverages probabilistic probe selection. With the unpredictability in random inspection, SpeedCam enables the effective detection of violation in bandwidth consumption. Through the evaluation with a proof-of-concept implementation of SpeedCam, we have demonstrated that our probabilistic approach can effectively diagnose multipath flows. In addition, we have further discussed a possible improvement in the probe selection by leveraging the control plane information. We anticipate that this work can contribute to various network management approaches for multipath enabled environments.

Acknowledgement

We would like to thank all our colleagues, project partners, and especially the SCIONLab team for their valuable input and feedback to this paper. We gratefully acknowledge support from ETH Zurich, and from the Zurich Information Security and Privacy Center (ZISC).

References

[1] W. Xu and J. Rexford, "Multi-path interdomain routing," in *ACM SIGCOMM*, 2006.

[2] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? Designing and implementing a deployable multipath TCP," in *USENIX NSDI*, 2012.

[3] O. Bonaventure, C. Paasch, and G. Detal, "Use cases and operational experience with multipath TCP," IETF RFC 8041, https://datatracker.ietf.org/doc/rfc8041/, (2017).

[4] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The QUIC transport protocol: Design and internet-scale deployment," in *ACM SIGCOMM*, 2017.

[5] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush, "From Paris to Tokyo: On the suitability of ping to measure latency," in *ACM IMC*, 2013.

[6] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," *IEEE/ACM ToN*, vol. 25, no. 2, 2017.

[7] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song, "FlowQoS: QoS for the rest of us," in *HotSDN*, 2014.

[8] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM CCR*, vol. 34, no. 4, 2004.

[9] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen, "cSamp: A system for network-wide flow monitoring." in *USENIX NSDI*, 2008.

[10] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM CCR*, vol. 36, no. 2, 2006.

[11] J. Kwon, J. A. García-Pardo, M. Legner, F. Wirz, M. Frei, D. Hausheer, and A. Perrig, "SCIONLab: A next-generation Internet testbed," in *IEEE ICNP*, 2020.

[12] V. Mohan, Y. J. Reddy, and K. Kalpana, "Active and passive network measurements: A survey," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, 2011.

[13] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," in *IEEE INFOCOM*, 2003.

[14] C. Chaudet, E. Fleury, I. G. Lassous, H. Rivano, and M.-E. Voge, "Optimal positioning of active and passive monitoring devices," in *ACM CoNEXT*, 2005.

[15] B.-Y. Choi, J. Park, and Z.-L. Zhang, "Adaptive packet sampling for accurate and scalable flow measurement," in *IEEE GLOBECOM*, 2004.

[16] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network monitoring in software-defined networking: A review," *IEEE Systems Journal*, vol. 12, no. 4, 2018.

[17] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "RFC 1157: Simple network management protocol (SNMP)," 1990.

[18] C. Lonvick, "RFC 3164: The BSD syslog protocol," *Request for Comments, IETF*, 2001.

[19] H. Adhari, T. Dreibholz, and M. Becke, "SCTP socket API extensions for concurrent multipath transfer," IETF Internet Draft, https://datatracker.ietf.org/doc/draft-dreibholz-tsvwg-sctpsocket-multipath/, 2018.

[20] S. Dawkins, "Path aware networking: A bestiary of roads not taken," IETF Internet Draft, https://datatracker.ietf.org/doc/draft-dawkins-panrg-what-not-to-do/, 2018.

[21] B. Trammell, J.-P. Smith, and A. Perrig, "Adding path awareness to the Internet architecture," *IEEE Internet Computing*, vol. 22, no. 2, 2018.

[22] K. Gaertner, J. Kwon, and D. Hausheer, "Speedcam: An efficient flow monitoring approach for multipath communication," OVGU Magdeburg, Tech. Rep. NetSys-TR-2021-01, 2021. [Online]. Available: http://www.netsys.ovgu.de/netsys_media/publications/NetSys_TR_2021_01.pdf