Path Oracle: Improving Performance of Path-Aware Applications in SCION

Thorben Krüger
Otto-von-Guericke University
Magdeburg, Germany
thorben.krueger@ovgu.de

Clemens Beck
Otto-von-Guericke University
Magdeburg, Germany
clemens.beck97@gmail.com

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

***Index Terms*—Path-awareness, Path Oracle, SCION**

***Abstract*—SCION is a next-generation Internet architecture enabling path-awareness for end-host applications. Contrary to today's single-path Internet, where paths are chosen based on BGP, path-awareness is a novel property allowing applications to select paths for their traffic based on specific goals, e.g. maximizing the bandwidth or minimizing the latency. To this end, SCION offers a list of path alternatives along with static path properties based on which SCION end hosts can select their paths. However, since the properties of these paths are often dynamically changing, SCION end hosts also need to rely on self-performed probes for their path selection. Such probes are usually time consuming and create a lot of overhead.**

In this paper, we therefore design and implement a system for sharing dynamic path properties in SCION-based networks, enabling applications to select paths in a more informed manner. Our implemented Path Oracle derives to-be-expected performance metrics of paths from donated end-to-end path performance data. For this purpose, the Path Oracle offers two interfaces to applications, one for querying path scores and one for donating measured metrics.

The Path Oracle tracks the currently achievable throughput on paths by maintaining a network view on a link level, enabling to derive scores for paths without previously received data donations. The service incorporates donated measurements of applications performing network-bound input or output into its network view. The implemented Path Oracle and throughput service were evaluated in the global SCIONLab test network. Here the Path Oracle was able to successfully determine paths offering the best throughput. In addition we conducted further experiments to evaluate the system under a range of different simulated network configurations and high load scenarios, achieving significant performance gains.

I. Introduction

Every day we use a variety of Internet applications with different networking requirements. For example, Internet telephony requires low latency for a good user experience; meanwhile, the focus for downloading a large file is on high throughput. Those properties are highly dependent on the underlying path over which packets are forwarded.

In today's Internet those paths are chosen by ISPs based on the Border Gateway Protocol (BGP). While traffic usually flows over single paths, routes may change suddenly, possibly leading to unexpected changes in latency, throughput or, at worst, even broken connections. Moreover, neither the receiver nor sender of a packet can influence the path of received or sent packets. Applications might be limited to suboptimal paths for their communication requirements, unable to easily avoid bottlenecks on those paths.

Modern Internet architectures like SCION [1] offer path-awareness and allow endpoints to select the path of their packets which is ensured cryptographically. Thereby, applications can select a path matching their requirements, possibly improving the application's performance significantly. Path properties can be either static or dynamic. A static property is valid for the lifetime of a path, e.g., the maximum bandwidth or the number of hops. Meanwhile, dynamic properties, e.g., current throughput or latency, change continuously.

SCION offers an extension for configuring and distributing static properties, which are available for endpoints during path selection. However, currently endpoints have limited possibilities to gain insight into the dynamic properties of paths without actively benchmarking those. As those benchmarks consume time and computing resources, static properties are often used as the only basis for path selection. However, since static properties do not take the load of the infrastructure into account, their value is limited, and a selected path can be far from optimal. For example, a path with powerful hardware under high load can perform worse than a path with weaker hardware experiencing low load.

To this end, our paper proposes a Path Oracle service aggregating path metrics from connections to derive dynamic path properties. Those properties grant applications insight into the underlying network and the currently achievable performance of different paths. As a result, applications are able to select paths in line with their requirements.

The remainder of this paper is organized as follows: Section II reviews related work and provides necessary background, before our main Path Oracle approach is presented in Section III. We experimentally evaluate its performance in Section IV. Finally, Section V summarizes our contribution.

II. Background and Related Work

SCION is a next-generation Internet architecture already seeing real-world deployment [2], which is designed to provide *path awareness* to end-hosts, allowing applications to improve their performance by selecting suitable paths based on their networking requirements. A detailed overview on the SCION architecture is provided in the SCION book [1]. In this section, we give a brief summary of the related work relevant to our chosen approach. For more detailed background information please refer to [3, p. 10].

In today's Internet, applications have little to no information about their underlying network topologies. As a result, applications, such as peer-to-peer filesharing or content delivery network clients, which have a choice between a number of ostensibly equivalent remote endpoints, may not always pick the best candidates according to current network conditions or geographical distance.

The Application-Layer Traffic Optimization (ALTO) protocol (see RFC7285 [4]) allows applications to obtain costs or other network metrics to remote endpoints. Most importantly, applications can query ALTO services with the ALTO protocol and make a more sophisticated choice on which endpoints to connect. As a result, applications can benefit from better performance, and the whole network infrastructure can achieve better utilization. The protocol specification focuses on disseminating information about routing costs but could also support more dynamic metrics. However, as SCION hosts have full control over path(s) to the desired endpoint, the ALTO protocol with its narrower focus on endpoint selection is currently not an appropriate framework to improve path-selection in a path-aware environment like SCION.

Moreover, SCION already makes static path properties like the maximum bandwidth or minimum latency of paths transparent to endpoints. However, due to their *static* nature, these properties inherently cannot reflect *dynamic* network or system load and their utility for path selection is therefore limited.

In contrast, our proposed Path Oracle aims to distribute dynamic network properties based on an up-to-date network view. In our own previous paper on PANAPI [5], we have already sketched an approach that allows assembling a network view from host perspective. The Path Oracle extends this work by moving the process to a more privileged vantage point at the network level.

III. Path Oracle

Our concept of a Path Oracle for SCION-based applications (see Figure 1 relies on voluntary reports about observed end-to-end metrics from applications that have monitored the performance of their used paths. The Path Oracle aggregates this information and can be queried by endpoints for path quality scores that can be used for path selection. By design, the Path Oracle can support several types of scoring mechanisms for different path properties, serving a range of use-cases.

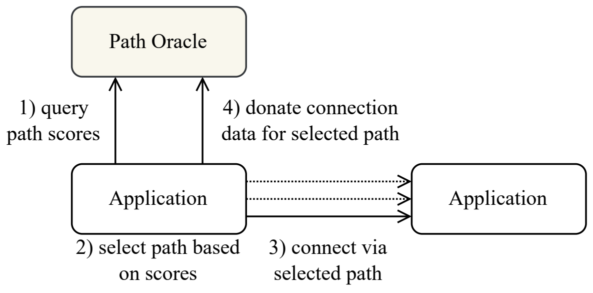
Path Oracle
1) query path scores
4) donate connection data for selected path
Application
Application
2) select path based on scores
3) connect via selected path

Fig. 1. Basic concept of the Path Oracle

A. System Design

Applications interact with a Path Oracle as part of their path-selection strategy. As its clients, they receive path quality information and are encouraged to monitor and report the performance on their chosen paths back to the Oracle.

The central component of a Path Oracle is its set of Scoring Services. Each Scoring Service maintains an internal Network View to derive scores for one path property, e.g., throughput or latency, from received data reports. A Path Oracle offers two APIs:

1) Reporting API: Accepts reports containing observed network performance metrics to be processed by the various Scoring Services. Each Scoring Service can define individual filters and requirements for reports.Reports matching all criteria get included in the Network View.

2) Scoring API: Allows Oracle clients to fetch path scores for destination Autonomous Systems (ASes). The Scoring API forwards requests to the Scoring Service, which processes the state of the Network View into path scores. After an Oracle client retrieves path scorings, those scores may get included in the path selection process by directly selecting paths with good scores or combining them with other properties.

B. Constraints and Limitations

The Path Oracle and its clients rely on a previously established trust relationship. This trust relation is crucial since the Path Oracle cannot validate incoming reports for correctness. Even without malicious intent, measurement errors are to be expected and Scoring Services will need mechanisms to detect and discard implausible reports.

Furthermore, Oracle clients must be convinced of quality and utility of the Oracle's recommendations while remaining aware of the limits of its "wisdom", which ultimately depends on the quality of the collected data.

The Path Oracle is designed to derive scores for SCION paths based on information gathered from end-to-end measurements. However, a complete end-to-end path includes, besides the path-aware inter-domain SCION path, non-path-aware intra-domain sections. For this reason, it is possible that endpoints in the same AS could experience different performance connecting to the same endpoint in a remote AS. Previous studies have shown that up to 40% of all bottlenecks of end-to-end paths occur inside ASes [6]. Thus, when an endpoint repeatedly reports worse performance on a path than the majority of other endpoints, it is probably bottlenecked at intra-AS-level its reports must be treated accordingly. For now, we assume all hosts inside an AS have equal connectivity to a given SCION border router.

C. Throughput Scoring Service

The Throughput Scoring Service can serve as a model for other possible Scoring Services. This service approximates the currently achievable throughput of a given path, defined as a path's theoretical available capacity. Especially applications that perform network-bound I/O are expected to benefit from using paths with high throughput.

The metric throughput was chosen as it heavily depends on the path's load. Furthermore, measuring the throughput of all available paths is more costly than measuring, e.g., the latency, making it unattractive for applications to probe themselves, especially for short-lived connections.

The Throughput Scoring Service accepts reports from applications that:

- performed network-bound I/O
- monitored the achieved throughput value

Plausible reports are included in an internal Network View, which is maintained on both a link and a path level. The link-level view is a graph containing all known interfaces as nodes and all known links as edges. Newly received reports get associated with all links of the report's path. Meanwhile, the path-level view maps all paths to an Exponential Moving Average (EMA). The EMA is initialized with the throughput of the first report of the corresponding path and gets updated with every newly donated throughput tp_n using a smoothing factor s between 0 and 1.

$$EMA_0 = tp_0$$
$$EMA_n = EMA_{n-1} * (1 - s) + tp_n * s$$

The path EMA estimates a path's throughput and not equivalent to the path's score. A path score is defined as the minimum over all link scores on the path. The link score itself is another EMA based on the throughput of the filtered list of the reports associated with the link. Not all reports are included in the link's score, as only a few paths are bound to the link's performance and are instead bottlenecked on another link. To this end, a *bottleneck ratio* is employed for each link, dropping all paths with relatively low path EMAs from consideration for the respective link EMAs.

With the link and path EMA, more relevant recent reports are weighted higher than older reports.

Overall, the whole procedure with its link-level and path-level processing ultimately allows deriving path scores even for yet unseen end-to-end paths without dedicated reports, as long as scores are available for some of its links.

D. Oracle Client

A widely used library for SCION networking is *pan* [7], which offers abstractions that allow software developers to implement their custom path selection strategies. We focus on the *Throughput Oracle Path Selector* which periodically fetches scores from the Path Oracle to select the path with the highest throughput.

For QUIC-based connections, the Oracle Client uses the *quic-go* [8] library underneath, which supports connection monitoring via callbacks into a tracer module. Implementing a custom tracer allows us to collect metrics for a connection, which form the basis for the client's reports to the Path Oracle. The report gets donated to the Path Oracle each time a path is switched or after a configurable time frame expires. A report always includes monitored data for exactly one path.

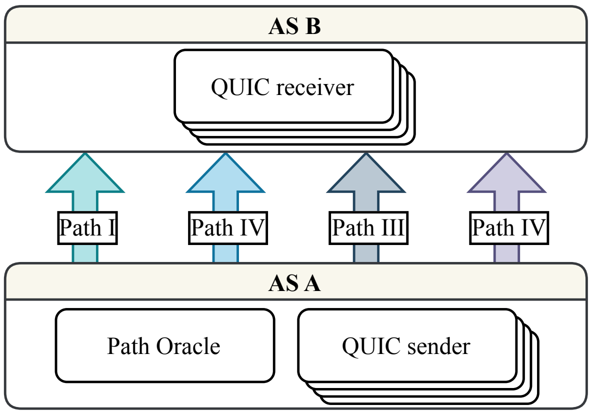

Fig. 2. Basic parent-child topology, forming the basis for the load-distribution experiments

IV. Experiments & Evaluation

Several experiments were performed to evaluate the Path Oracle, both in a real-world setting as well es in a more controlled environment with multiple concurrently active applications, regarding path load distribution.

A. Experiment in the SCIONLab network

An initial experiment was conducted to evaluate Path Oracle performance in the SCIONLab [9] test lab environment. This experiment showed that the

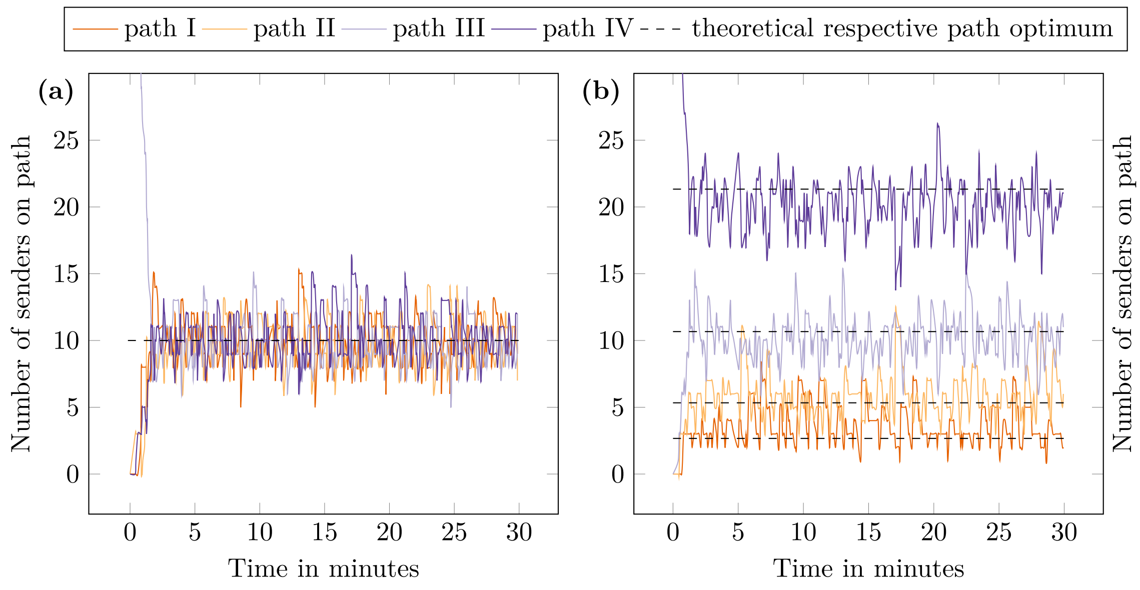

Fig. 3. **Path Distribution via Path Oracle** (40 senders): **(a)** Network topology with equal path bandwidths. **(b)** Network topology with double the bandwidth of path n on path $n+1$.

Path Oracle accurately derives the throughput of paths from its received donations. Clients using those scores can expect good performance compared to a random path selection. Nevertheless, in this case, the Oracle-based path selection strategy did not result in a measurable improvement against the baseline strategy of selecting the shortest paths. (Plot not shown for lack of space.)

B. *Experiment with Paths of equal capacity*

The following experiment deploys the topology of Figure 2 in a simulated environment. The two ASes are connected via four paths with equal capacity (32 Mbit/s). In a scenario where multiple applications perform network-bound I/O concurrently from one AS, the optimal distribution from a network point-of-view would be a balanced use of the paths. In a real-world scenario, e.g., where each application uploads a file of the same size, this would equalize the required upload time.

Without insight into the network's load and dynamic path properties, path selectors cannot perform an adequate path selection in this topology.

For evaluating the distribution where applications use dynamic throughput property queried from the Path Oracle, 40 QUIC senders and receivers are deployed in the network. Each sender uses the Throughput Oracle Path Selector. Before each experiment, the Path Oracle network view is bootstrapped with initial data for each path.

As can be seen in Figure 3(a), all senders initially pick the path which happens to have the highest score at first. After a short delay, the client distribution fluctuates around the expected optimum. The fluctuation is caused by senders switching paths based on their periodic Path Oracle consultations. Each switch releases bandwidth on the previous path and occupies throughput on the new path. These changes are reflected in the subsequent reports of senders having this path selected. These changes are reflected in updated scores and the distribution is re-balanced. Despite this effect, the results are close to the optimum and promise a better load distribution for real-world deployments.

C. *Experiment with Paths of different capacity*

Under the same experimental setup with a different path configuration, the circumstances are even harder to handle without the use of dynamic path properties. Here, we double the capacity from path to path, i.e., path I has 8Mbit/s, path II has 16Mbit/s, path III has 32Mbit/s and path IV has 64Mbit/s for a total of 120Mbit/s.

Purely from an analytical perspective, the optimal distribution of senders would be the following:

- $6.\overline{6}\%$ of senders via path I (at 8Mbit/s)
- $13.\overline{3}\%$ of senders via path II (at 16Mbit/s)
- $26.\overline{6}\%$ of senders via path III (at 32Mbit/s)
- $53.\overline{3}\%$ of senders via path IV (at 64Mbit/s)

When all participants naively use the path with the best static bandwidth, path IV, which promises the maximum bandwidth, would be overused. Conversely, the paths I, II and III would be underused. A /textitrandom path selection would at best distribute applications evenly, which would result in only some participants achieving high throughput (because they randomly pick a good path).

Again, a bootstrapped Path Oracle, 40 QUIC senders, and receivers are spawned. Similar to the previous setup, all senders start using one path initially, as shown in Figure 3(b). Shortly after, the applications get distributed on the other paths. Again, similar to the previous setup, the distribution fluctuates around the optimum.

These experiments show that the distribution of applications on different paths can be optimized by utilizing the Path Oracle. The applications are automatically distributed proportionally according to the respective capacity available on a path.

V. Summary and Conclusion

Applications have different requirements and could improve their performance if they could select a path matching their demands. Given an Internet architecture like SCION that (contrary to the current Internet) allows for host-based path selection, to make an informed path choice, a mechanism to determine path properties is required. To address this, we propose the Path Oracle. We designed and implemented a prototype which aggregates path metrics and assembles an internal network view to derive path scores for SCION paths.

The Path Oracle is a donation-based system that derives path scores from donated reports from applications. The reports contain passive measurements of end-to-end traffic. Thus, the application's networking layer is aware of its requirements enabling the path selection mechanism to adapt.

The architecture of the Path Oracle is sufficiently flexible to support several scoring mechanisms and future extensions. We implemented a first mechanism to rank the achievable throughput on several possible paths. This service processes incoming reports in a way that allows it to also derive scores for paths for which no donation was ever received.

With our experiments, we could show that path selection strategies that do not utilize dynamic path properties lack the insight to perform a sophisticated path selection. In these scenarios, we could show that the Path Oracle can derive scores that ultimately successfully load balance applications onto different paths, providing improved throughput over random or shortest path selection.

Concluding, the Path Oracle promises to allow applications a more sophisticated path selection. For now, especially network-bound I/O performing applications can benefit from the throughput metric offered by the Path Oracle. Thereby, with the continuous advancement of the SCION ecosystem, the Path Oracle can be evaluated for additional path metrics and different use case scenarios.

VI. Future Work

It is an obvious next step to design further Scoring Services targeted at other path properties that may be desirable, such as latency minimization.

Further (and deeper) integration of the oracle mechanism into networking libraries and APIs [10] would allow a wide range of applications to benefit. Such integration could also benefit the approach as a whole, as the acceptance and the expected number of donated reports rises with broader integration into libraries. This would in turn increase the availability of helpful metadata reflecting a useful cross-section of application requirements.

Of course, care must be taken to meet privacy concerns (and legislation). Too detailed insight into the application's traffic could cause a Path Oracle to inadvertently leak sensitive information about user behavior. Therefore, the right balance needs to be found between beneficial data mining and prudent data economy.

References

[1] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig, The Complete Guide to SCION. Springer, 2022.

[2] C. Krähenbühl, S. Tabaeiaghdaei, C. Gloor, J. Kwon, A. Perrig, D. Hausheer, and D. Roos, "Deployment and scalability of an inter-domain multi-path routing infrastructure," in Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies, 2021, pp. 126–140.

[3] C. Beck, "Improving the Performance of Path-Aware Applications by Utilizing Dynamic Path Metadata in SCION with a Path Oracle," Master's thesis, OVGU Magdeburg, Germany, 2022, https://www.netsys.ovgu.de/netsys_media/Downloads/ThesisBeck2022.pdf.

[4] "RFC 7285 - Application-Layer Traffic Optimization (ALTO) Protocol," 2014.

[5] T. Krüger and D. Hausheer, "Towards an API for the Path-Aware Internet," in Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration, ser. NAI'21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 68–72.

[6] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating Internet Bottlenecks: Algorithms, Measurements, and Implications," SIGCOMM CCR, vol. 34, pp. 41–54, 2004.

[7] "Path aware network (PAN) library," 2022, https://pkg.go.dev/github.com/netsec-ethz/scion-apps/pkg/pan.

[8] "quic-go documentation," 2022, https://pkg.go.dev/github.com/lucas-clemente/quic-go.

[9] 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. [Online]. Available: https://netsec.ethz.ch/publications/papers/icnp2020_scionlab.pdf

[10] B. Trammell, M. Welzl, R. Enghardt, G. Fairhurst, M. Kühlewind, C. Perkins, P. S. Tiesel, and T. Pauly, "An Abstract Application Layer Interface to Transport Services," https://datatracker.ietf.org/doc/draft-ietf-taps-interface/15/.