LARS: Keeping Local Traffic Local with Latency-Aware Route Servers at IXPs

David de Andres Hernandez
IMDEA Networks Institute, UC3M
david.deandres@imdea.org

Yasin Alhamwy
University of Kassel
alhamwy@uni-kassel.de

Daniel Wagner
DE-CIX, MPI-INF
daniel.wagner@de-cix.net

Maximilian Stephan
Technische Universität München
maximilian.stephan@tum.de

Matthias Wichtlhuber
DE-CIX
matthias.wichtlhuber@de-cix.net

Oliver Hohlfeld
University of Kassel
oliver.hohlfeld@uni-kassel.de

Abstract

Internet Exchange Points facilitate traffic exchange through Route Servers that use BGP for path selection without considering latency. This paper presents LARS, the first system to enable latency-aware path optimization at IXP route servers. It requires no changes to BGP and no bilateral coordination among members, thus directly enabling latency-aware routing for all IXP members. LARS continuously probes the route-server RIB, annotates routes with RTT-derived BGP communities, and enables members to *(i)* enforce a configurable latency radius that suppresses high-latency routes and *(ii)* prefer lower-latency alternatives when multiple paths exist. We evaluate LARS at a very large IXP using a production-level prototype that demonstrates stable latency estimation, effective tail-latency control via selective redistribution, and potential to optimize BGP best paths.

CCS Concepts

• **Networks** → **Routing protocols**; **Network measurement**.

Keywords

IXP, BGP, Route Optimization, Latency Optimization, Peering

ACM Reference Format:
David de Andres Hernandez, Yasin Alhamwy, Daniel Wagner, Maximilian Stephan, Matthias Wichtlhuber, and Oliver Hohlfeld. 2026. LARS: Keeping Local Traffic Local with Latency-Aware Route Servers at IXPs. In *ACM SIGCOMM 2026 Conference (SIGCOMM '26), August 17–21, 2026, Denver, CO, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3789240.3829211

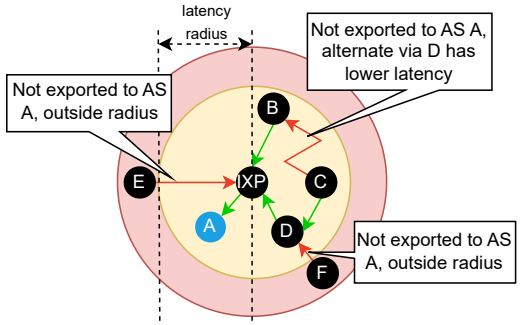


Figure 1: LARS implements two ideas to optimize routes member AS A receives from an IXP's route server: *(i)* AS A can define a custom latency radius beyond which it does not want to receive routes, neither from other peers (AS E) or from peers' customer cones (AS F via AS D); *(ii)* within the latency radius, LARS can optimize alternate paths for AS A: e.g., to prefer a lower latency path via AS B then C or D.

1 Introduction

Since the commercialization of Internet connectivity, the Border Gateway Protocol (BGP) path selection has reflected a persistent tension between engineering objectives and economic incentives. BGP's decision process explicitly prioritizes policy over performance, allowing operators to encode business relationships such as customer-provider and settlement-free peering. As a result, routing decisions often favor commercial preferences over end-to-end path quality, shaping today's Internet routing ecosystem. Consequently, systematic inefficiencies can arise when performance metrics such as latency are not explicitly considered, *even if they could be optimized within the given economic guardrails.*

Over the years, proposals have suggested incorporating performance awareness into interdomain routing [13, 28], including but not limited to active measurement-based selection [9, 25] and controller-based approaches [28, 32]. Despite this appeal, deployment has been limited and largely confined to hyperscalers at the edge [28, 32]. Reasons include limited benefits [10], operational complexity [32], scalability/stability concerns [27], and the absence of deployable solutions that fit existing operational practices without requiring bilateral coordination [13]. A broadly deployable approach to performance-aware routing in the Internet core does not yet exist, a gap we aim to close.

At the same time, Internet Exchange Points (IXPs) have become increasingly central to the Internet's topology. Modern IXPs show sustained growth in both member AS count and exchanged traffic volume. According to ISOC's IXP tracker, the number of IXPs with more than three members has doubled globally since 2019 [1]. This growth introduces new engineering challenges. Many large IXPs now span metropolitan areas [16] or even continents using distributed switching [15], which can introduce substantial latency differences between participants [22]. From BGP's perspective, all

SIGCOMM '26, Denver, CO, USA

peers connected to an IXP appear to be a single hop away, masking these latency variations and potentially leading to suboptimal paths [22]. IXPs thus open a so far *unused opportunity* to bring latency-aware routing to the Internet core at scale: a single IXP deployment can expose latency-optimized path choices to hundreds of member ASes without protocol changes or bilateral coordination.

In this paper, we present LARS, a system for centralized latency-aware path optimization at IXP Route Servers (RSes). LARS leverages the unique position of RSes, which already aggregate and distribute routing information for many member ASes, to enable latency-informed routing policies without requiring changes to BGP itself. The two main ideas of LARS are shown in Fig. 1: LARS allows IXP members to *(i)* define a configurable latency radius around the RS, beyond which they do not wish to receive routes (even over multiple AS hops), and *(ii)* optimize alternative paths in that radius by preferring lower-latency paths in BGP's best-path selection.

We evaluate LARS in practice at a very large IXP and demonstrate that such a system can be integrated into existing RS infrastructures with low effort, offering the IXP operator community a practical path to democratize latency optimization, particularly for smaller networks that lack the resources to deploy performance-aware routing solutions. Our contributions are as follows:

(1) We present the first large-scale characterization of route diversity and (high) geographic latency exposure at IXP RSes using global looking glass data, showing that roughly 40% of prefixes at large RSes have multiple, widely visible paths—enabling path optimization at scale—and that geographically remote prefixes are common at RSes (§ 2).
(2) We design LARS, a scalable, latency-aware route server architecture that measures path RTTs and encodes them using standard BGP communities, enabling members to enforce latency radii and prefer lower-latency alternatives without changes to BGP or coordination between endpoints (§ 3).
(3) We implement LARS in the production RS toolchain of a very large IXP. Using tens of billions of RTT samples, we show that RTT estimates are highly stable (≤1% variation for the vast majority of paths) and converge within operationally realistic timeframes (§ 4).
(4) We evaluate the benefits and costs of LARS, showing that filtering long-latency routes (>200ms RTT) has minimal traffic impact, while latency-aware path selection yields >10% RTT (up to 60ms) improvements for ≈5% of prefixes, with negligible RS overhead (§ 4.7-§ 4.9).
(5) We release our path latency measurement data at [18] and our ZMap extensions to measure path latencies at [2]: https://github.com/ds-ukassel/zmap-multipath-rtt.

2 Motivating BGP Optimizations

We begin by empirically motivating the premise of LARS, demonstrating that two key requirements are met: *i)* path diversity exists for path optimization, and *ii)* geographically remote prefixes are present at nearly any IXP for tail-latency optimization. We base this view on comprehensive global IXP looking-glass (LG) data. Notably, not all data can be used for every evaluation: IXPs differ widely, as do their LG deployments, necessitating manual data curation for certain analyses. We thus report the sample sizes used in each plot.

2.1 Approach: Studying BGP Routes at IXPs

Alice LG. We scrape the Alice LGs of 34 IXPs [8], which operate a total of 790 route servers (RSes) across all continents except Antarctica (see § C for the full list). This allows us to gain a comprehensive understanding of their members and the routes that each member receives. Our scraper [8] collects routing information from each LG, enabling us to build a global mapping of IXP members and route visibility—at all IXP sizes.

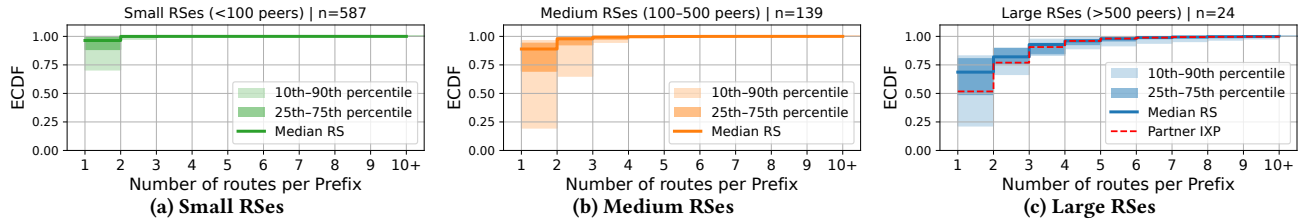
Ethical Considerations. We only collect public data exposed via LGs at a low request rate (one snapshot/day). We coordinated with the LG operations team at one of the largest IXPs in our dataset to find an acceptable scraping rate. The load at other (smaller) IXPs is comparable or lower.

Challenge: Per-Member View and Monitoring ASes. The RS master table obtained via an LG server does not directly represent the per-member view, i.e., whether an IXP member receives the route. This is because members can control the redistribution of advertised prefixes via a rich set of *action communities* to define routing policies. More precisely, many IXPs support selective redistribution of routes among participating members based on the presence of specific communities. As a result, these mechanisms can induce a large number of distinct routing views within a single RS, potentially as many as there are peers connected to it. Moreover, the mapping between communities and their associated actions varies significantly across IXPs, challenging a broad analysis. In addition, we find ASes in the LG data skewing the results (e.g., monitoring ASes from the IXPs).

Solution: Implementing Route-Server Logic and AS Grooming. We address the member view problem by reverse-engineering the redistribution logic for 4 IXP operators (31 RSes in total) using the IXP's documentation and emulating the member view based on community information in the RS tables. We cross-check our approach using per-member export route counts from the Alice LG API. Unlike these counts (simple counters on the RS), the *exported routes* per member cannot be scraped via the API (as they are usually not stored on the RS) and must be inferred as described above. To avoid skew, we manually exclude ASes that never affect routing at the IXP (e.g., connected only for LG visibility) and pure monitoring ASes (e.g., collectors with full exports).

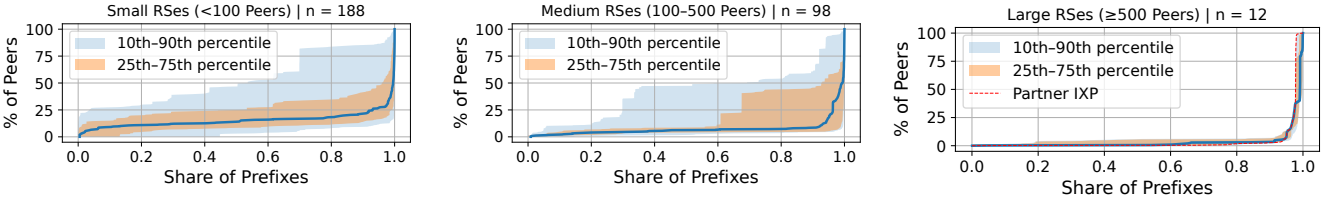
Finding Local Peers with Geolocation. To identify which prefixes are in geographical proximity to each IXP and define a latency radius with LARS, we enrich all prefixes at any RS with geolocation data provided by IPInfo. The geolocation data is measured using a large, distributed, and global probe network (see [17] p. 6 for details on the methodology). IPinfo achieves higher accuracy than alternatives such as MaxMind and community approaches using RIPE Atlas, with 89% of targets within 40 km compared to 73% and 55%, respectively, as reported by an independent paper benchmarking geolocation methods against a known ground truth [17].

View on Smallest Routable Units by Expanding to /24. We project all BGP routes onto IPv4 /24 address blocks, the smallest globally routable unit, and perform all geolocation analysis on this representation. Since IP geolocation is address-centric and individual BGP prefixes may span multiple physical locations, /24-level projection enables attribution at the most meaningful operational granularity.



(a) Small RSes (b) Medium RSes (c) Large RSes

Figure 2: Path diversity exists at IXPs. The bigger the RS, the more path diversity it sees. Sample size is indicated by n.



(a) Small RSes (b) Medium RSes (c) Large RSes

Figure 3: Most peers receive most routes, irrespective of IXP RS size. Sample size is indicated by n.

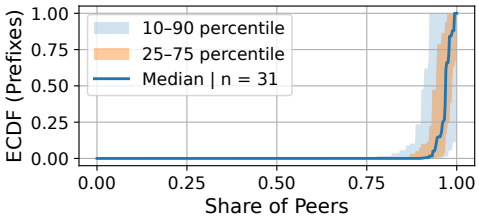


Figure 4: Share of peers to which each path-diverse prefix is distributed across 31 RSes of varying size. The majority of peers (≈90%) receive all path-diverse prefixes.

2.2 Result: Path Diversity Exists

Latency-driven route optimization is only possible when there is route diversity, i.e., multiple routes to the same prefix.

RS view. We demonstrate this presence by evaluating the route diversity at all studied RSes. The empirical CDF (ECDF)[1] in Figure 2 shows the number of available routes for small (< 100), medium (100-500), and large (> 500 peers) IXP RSes without taking selective redistribution into account. The figure shows two aspects. First, route diversity exists. At large RSes, ≈40% of the routes have alternate paths. As the percentile bands indicate, medium and small RSes can also exhibit high shares of alternate paths, but these shares are less predictable. Second, route diversity grows with RS size, offering more potential at larger IXPs.

Peer view. While Figure 2 examines path diversity among all paths available at each RS, we next study the routes exported to the IXP members (peers). We first look at how many exported routes each peer sees. Figure 3 shows the ECDF of the share of prefixes received by peers at the IXPs. The data is directly retrieved from the Alice LG API of all RSes, excluding: *(i)* those with <10 peers, *(ii)* IX.br due to measurement artifacts, and *(iii)* those that were unreachable at the time of the request. The prefix share was normalized to the peer that has the highest exported route count. We observe that, regardless of the RS size, most peers see most of the exported routes.

[1]We will also use the term CCDF hereafter to refer to the complementary ECDF.

Next, we investigate the distribution of path-diverse prefixes seen by peers. We investigate this by applying the selective redistribution logic via action communities to obtain the per-peer views of the 4 IXP operators with 31 RSes that provided the necessary public documentation. The RSes vary in size and are located at IXPs across four continents: Europe, Asia, North and South America. Figure 4 shows the share of peers to which each prefix with path diversity is distributed. Approximately 90% of peers receive all path-diverse prefixes.

Takeaways. Path diversity exists at IXPs: at large RSes, ≈40% of routes have alternate paths; smaller shares are found in small and medium RSes. Selective redistribution via action communities does not significantly affect the availability of diverse paths: 9 out of 10 peers could benefit from route optimizations by LARS.

2.3 Result: Tail-Latency Optimization

LARS enables IXP members to perform tail-latency optimization by allowing them to receive only routes whose measured latency is below a configurable threshold. This enables ASes to perform latency optimization by avoiding high-latency routes, e.g., useful for anycast networks or networks multi-homed at multiple IXPs. This can be beneficial, since all routes at IXPs are just one AS hop away, regardless of the latency required to reach them.

Footprint of IXPs/RSes. To study which potential for tail-latency optimization exists, we investigate the geographic distance of *all* announced prefixes to their respective IXPs/RSes. Therefore, we geolocate each prefix (split into /24s, see § 2.1) using IPInfo data, and each IXP/RS using public documentation. The median share of /24s that could be geolocated per IXP is 70.14%, and we successfully geolocated 171 IXPs/RSes at the metro-area level[2]. We then calculate the Haversine distance between IXP/RS and the prefix.

Figure 5 shows the ECDF of the distance between the different IXPs/RSes and the geolocation of the announced prefixes with the 25-75th and 10-90th percentile bands. We also show the estimated

[2]We also checked whether ungeolocated /24s cluster in specific network types, but found no obvious systematic bias.

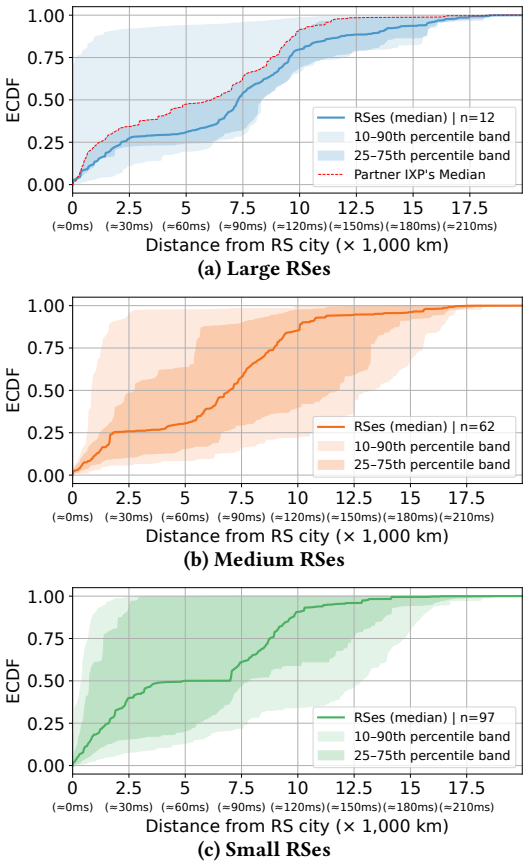


Figure 5: ECDF of prefix distances by category. Smaller IXP RSes display wide variation in remote prefixes, whereas larger IXP RSes show a consistent presence, reflected in the shrinking interquartile band.

RTT for the distances under the x-axis ticks, calculated from the speed of light in fiber with a line-of-sight path-stretch factor of 1.2 (see [12, 14]). We can make two observations. First, the plots show that larger IXPs/RSes generally have a much more international distribution of prefixes, whereas smaller IXPs/RSes *can* have more local networks. While not all of these prefixes are *currently used* by member ASes, they present selectable alternatives for LARS. Second, a notable share of the address space is far from the IXPs/RSes, and reaching it would incur high latency. Depending on their operational needs, such high-latency routes can be removed by LARS (e.g., for anycast networks such as root DNS servers or hyperscalers to extend their current path optimization capabilities).

Takeaways. Tail-latency prefixes can be found at almost any IXP's RSes. Smaller and medium IXPs can have a more local footprint, but tail-latency prefixes cannot be safely excluded. Large IXPs are highly likely to distribute tail-latency prefixes at RSes, while appearing as a one-AS hop in BGP routing.

3 LARS System Design

An overview of the LARS system design is shown in Figure 6. We design LARS to give IXP members control over the latency properties

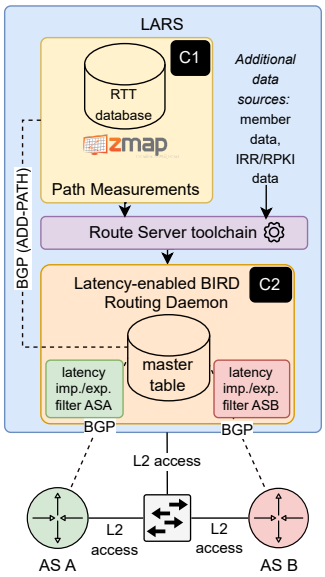


Figure 6: LARS system design and its major components.

of the routes they receive from a route server. Unlike existing mechanisms measuring the RTT to the *member's router*, LARS *measures RTTs from the route server to all routed hosts* (see Fig. 7), thereby *constructing a latency-aware view of each member's customer cone.*

A member can *(i)* restrict itself to routes whose measured latency falls below a configurable threshold, *(ii)* bias the BGP best-path decision toward lower-latency alternatives, or *(iii)* fall back to the standard BGP selection (no latency optimization). Achieving latency-aware routing at an IXP requires two capabilities, realized in two components (see Fig. 6): C_1, the continuous collection of path-latency measurements at the IXP, and C_2, an RS that can expose latency-optimal paths to members based on latency measurements. Both components interact by exchanging information: C_2 provides routing information to C_1, and C_2 incorporates path-latency information from C_1 into its routing decisions during the configuration generation process triggered by the RS toolchain, which controls every aspect of the RS operation. In the following, we describe the details of C_1 and C_2.

3.1 C_1: Active Path RTT Measurements

Component C_1 implements an active probing mechanism to measure RTTs over *all* available paths announced to the RS (see Figure 7 for an overview of the setup and integration into the IXP). The active measurement system is connected to the IXP's peering LAN. To receive all announced paths to the RS, and not only the selected best paths, it maintains an established ADD-PATH [31] enabled BGP session with the RS component C_2 (see Figure 6).

Adjustments to ZMap. To probe *all* paths announced to the IXP's RS, we extend ZMap [19]—a highly optimized scanning tool—with a multipath scanning module that lets LARS explicitly set the next-hop router's MAC address for each candidate path. By selecting the specific peering router for egress, we force traffic off the default route and can measure every possible path.

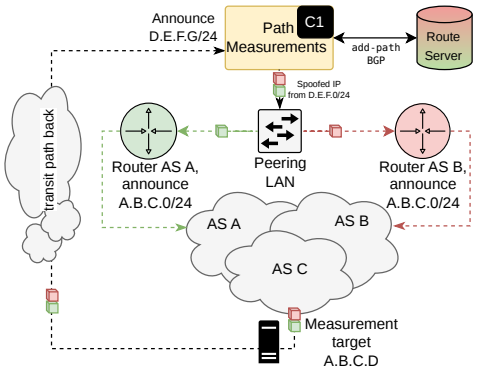


Figure 7: C_1 integrates at IXPs by *(i)* establishing an add-path-enabled BGP session with the RS to learn announced paths, and *(ii)* connecting to the peering LAN to send probes.

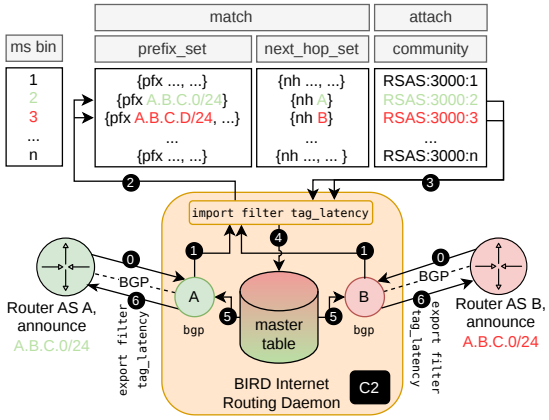


Figure 8: C_2 integration with IXPs. Each (prefix, next_hop) combination is mapped to a community value representing the latency in ms.

We implement multipath scanning for several ZMap probe modules rather than once at the ZMap core to avoid modifying and retesting core ZMap components. Our implementation [2] exploits ZMap's feature for sending multiple probes to each target host. We enforce the egress path by cycling over and setting all MAC addresses of the peers that have an announcement to the host. Therefore, we maintain a mapping of prefixes to available next-hop destination MAC addresses in memory as a hashmap. While these changes to ZMap seem straightforward, they are non-trivial as ZMap's architecture does not support this use case.

ZMap RTT Probe Modules. To generate round-trip time (RTT) samples, we overload protocol fields [11, 19] to encode the sending timestamp and the BGP path ID. This is required to correctly associate incoming responses with the corresponding egress path, i.e., we can assign latencies to the correct BGP announcement of the RS peer. For this, we consider four mechanisms: ICMP echo request messages, TCP-SYN packets, TCP-ACK packets, and UDP datagrams. They all leverage the fact that responsive addresses will respond with a packet whose header will include the encoded information. Details on the individual mechanisms and the encoding of the sending timestamp and path identifier are provided in Appendix A. Using these modules, released in [2], any multihomed network, e.g., research institutions, can perform similar studies.

Selecting a Probe Module. Because response rates vary across probe protocols (ICMP, UDP, or TCP), we evaluated them in the wild, i.e., over a random set of 1M targets. Detailed results of the probe selection experiments are presented in Appendix B. We select ICMP for its higher response rate and simplicity, since the latency distributions across the subset of targets that respond to all four probe types are statistically equivalent.

Response validation. To distinguish probe responses from background traffic, we rely on ZMap's default cryptographic validation mechanism, as all probes embed validation sequences. We discuss multipath response validation in § A.1, and investigate anomalous response patterns in § 4.4.

Transit address spoofing. While replies to our probes could, in theory, travel back to LARS' interface in the peering LAN, we spoof the source address to an IP address from a block we only announce at a secondary interface connected to a large, global transit provider. This forces all replies through a transit path back to our LARS instance. While not strictly necessary, we find this to have two beneficial effects: *(i)* it roughly doubles the response rate on a per IP basis, as more hosts have a route back to LARS, and *(ii)* if a good transit provider is chosen, we found to have stable routes back to LARS over long periods of time (see § 4.4).

3.2 C_2: Providing RTT-Aware Routes

LARS' second component (C_2) annotates BGP route advertisements with RTT information derived from path measurements (C_1). Figure 8 presents the component's overview and workflow.

Route Server 101. At an IXP, an RS is a central component that eases multilateral peering between members. The RS re-advertises member routes received via eBGP sessions to other members. At many IXPs, the RS stack is built upon the Bird Internet Routing Daemon (BIRD) [3], often complemented by automation such as IXP Manager [4]. To keep the RS configuration up to date with each member's intent, policies, and IRR and RPKI ROA information (used for route validation), IXP operators use automated toolchains that periodically rebuild the configuration based on updates submitted through the member management portal.

LARS at an IXP RS. We design C_2 to be compatible with IXP tool chains and to target a BIRD-based deployment, chosen for its widespread use. We assume that existing toolchains for RS configuration include all existing state-of-the-art BGP routing security mechanisms: bogon filtering, Internet Route Registry (IRR)-based filtering, and Resource Public Key Infrastructure (RPKI) origin validation. LARS merely adds the possibility of integrating the RTT information database acquired with C_1 throughout configuration generation.

RTT data aggregation. Throughout the configuration generation process, the toolchain aggregates measurements for single IPs to routes. It does so by first selecting the minimum RTT among all RTTs measured within a /24, and then aggregating the minima into larger routed prefixes by calculating the median of all /24s contained in the route. This method of aggregation is validated

through large-scale measurement campaigns at a large IXP (see Sec. 4.4-Sec. 4.6).

RTT community tagging on import. LARS attaches an inbound policy filter to every BGP session at the RS (❶ in Figure 8) and applies it to incoming route advertisements (❶). Before generating the RS configuration, the toolchain processes the RTT database and maps each measured RTT to a configurable RTT bin. Each RTT measurement is associated with the exact *(prefix, next-hop)* pair that was used during probing. Consequently, the toolchain compiles a lookup table that maps every measured *(prefix, next-hop)* pair to a single RTT bin and the corresponding BGP community encoding that RTT value.

For each RTT bin, the toolchain constructs two parallel match lists (❷) from measured prefix–next-hop pairs: *(i)* a `prefix_set` list whose element at index j contains the prefixes assigned to RTT bin j, and *(ii)* a `next_hop_set` list whose element at index j contains the next-hop addresses measured for those prefixes. The import filter `latency_tag` then implements this lookup as one rule per RTT bin. A rule only matches if both attributes of an incoming route match simultaneously: the advertised prefix must be contained in the bin's `prefix_set`, and the route's next hop must be contained in the corresponding `next_hop_set`. When both conditions hold, LARS has identified the *(prefix, next-hop)* pair and therefore attaches the BGP community corresponding to the RTT bin (❸).

The attached community encodes the measured RTT in milliseconds using the last two octets, allowing millisecond precision. After the RTT community has been attached, the route is stored in the `master` routing table (❹).

If a tagged route is withdrawn, the tag is removed from the `master` table along with the route. When a new route is announced, a measurement might not yet exist. Therefore, new routes receive no tags until a future scan yields a measurement and is pushed into the RS configuration by a regular toolchain run. These runs usually happen multiple times per day, depending on the IXP operator.

In effect, the import policy acts as a compiled lookup table that maps previously measured (prefix, nh) pairs to an RTT bin and its corresponding community.

Route export. From the `master` routing table, the tagged route gets redistributed to the peers' export table (❺), as implemented by the RS's remaining logic (e.g., selective redistribution to honor business relationships and preserve Gao-Rexford [21] constraints). Finally, the RS runs the well-known BGP decision process independently for each export table (❻). For route export, LARS offers two operation modes via the `export filter tag_latency`: passive and active.

Passive (RTT Tagging). In passive mode, RTT communities attached during import are preserved and exported unchanged. The RS does not alter the BGP decision process. The tags serve purely as informational metadata, enabling member ASes to incorporate RTT information into their own routing policies if desired.

To support use cases where members require visibility into multiple paths (not only the best), ADD-PATH-enabled sessions [31] are also supported. Some large IXPs already offer ADD-PATH support in their route servers (e.g., IX.br [7], MegaPort [6]). Enabling ADD-PATH support is generally straightforward: simply set a flag for each BGP session in the route server's configuration. This can be easily accomplished during the deployment of LARS.

Active (RTT-Based Best-Path Selection). In active mode, the RTT community tag attached during import is interpreted by the export policy relative to a user-configurable RTT radius—set by the IXP member in the customer portal—to derive a route-specific `LOCAL_PREF` (LP) value. By rewriting `LOCAL_PREF` based on the tag, latency information directly influences the BGP best-path selection process without requiring modifications to BIRD or the BGP protocol. The transformation is applied independently to each route during export. In a standard RS configuration (i.e., without LARS), all routes have `LOCAL_PREF = 100`. Since `LOCAL_PREF` is the first attribute evaluated in BGP best-path selection, adjusting it allows RTT-aware selection without modifying BIRD or the BGP protocol.

The export filter performs a mutually exclusive classification; for each route, exactly one case applies:

(1) **Tag > radius:** if the RTT value encoded in the community exceeds the member-defined latency radius, the route is suppressed. If this is the only available route for a prefix, the prefix becomes unreachable. This way, LARS gives operators the much-needed ability to trade route-set size for stricter latency bounds, thereby providing a tool to alleviate problematic long-latency routes. This allows multihomed networks (CDNs/Anycast/etc.) to optimize routing for local PoPs via RSes, which is currently challenging with classical traffic-engineering methods.
(2) **Tag ≤ radius:** if a RTT tag is present and within the radius, $LP = RTT_{\max} - RTT_{\text{tag}}$, where $RTT_{\max}$ is the maximum RTT bin and RTT_{tag} is the route RTT; lower RTT yields higher LP.
(3) **No tag:** if no measurement exists, a member-defined fallback LP (default $RTT_{\max}$) is assigned; values $> RTT_{\max}$ always prefer unmeasured routes, while values $< RTT_{\max}$ allow sufficiently low-RTT measured routes to outrank them. Notably, this case is an absolute corner case; if a host answers to our ICMP probes, it usually does so via all available paths.

We designed LARS to never introduce invalid routes but to allow *filtering* and *ranking* routes that IXP members receive anyway. All routes are imported into the route server RIB, and export filtering occurs only when peers actively set latency thresholds. All other members continue to receive the original routes. A key step to prevent invalid routes is to honor selective redistribution communities. This step is already implemented in the route server policies and not altered by LARS. In active mode, LARS only adjusts the `LOCAL_PREF` values of already received routes (default 100 for all routes at the IXP's RS). This way, LARS never introduces routes that IXP members would not receive in the absence of LARS, and thus, LARS never introduces routes that violate interconnection agreements.

Design Implication. The scheme remains fully compatible with existing BGP policy mechanisms and can be implemented entirely in BIRD's policy language without daemon modifications.

4 Integrating LARS into a Very Large IXP

To implement and evaluate LARS, we partner with the operator of a very large IXP (hereinafter, the "partner IXP"). We have access to the RS config generator toolchain and RIB snapshots, and we place our measurement system in the IXP's peering LAN to obtain latency information for all available paths at the IXP's RS. By integrating LARS into the RS config, we show that LARS incurs only a very small performance penalty, even at a very large RS. Further, our study

shows that LARS manages to achieve latency gains in an already highly optimized IXP environment. More importantly than absolute latency gains, LARS allows operators to eliminate problematic high-latency routes, as evidenced by its ability to reduce tail latency by up to 30%. Our evaluation thereby shows that LARS provides substantial gains in a single IXP deployment.

Each IXP deployment of LARS operates independently, requiring no coordination between IXPs. While our evaluation already demonstrates substantial gains from a single IXP deployment, and multiple IXPs appearing on the same path is operationally uncommon, if multiple IXPs on a path do deploy LARS, latency can only further improve, as each deployment independently filters and optimizes routes within its own radius.

4.1 Datasets

We collect public data as well as IXP-specific data by deploying LARS at our partner IXP. Next, we discuss the datasets used.

IXP RS: BGP Routing Data. We obtain access to the IXP's RS Routing Information Base (RIB). This RIB is non-filtered, i.e., community-based filtering or best-path selection have not been applied. This is in contrast to public BGP collector tables (e.g., from RIPE RIS) that contain only the best paths and thus enables us to compute the exact set of routes advertised to each IXP member. We remark that the IXP's route server filters invalid routes that are not advertised to members and are not in our RIB snapshot, e.g., based on IRR or RPKI ROA information. The RIB contains 397k distinct prefixes. Note that these contain all available paths prior to the best paths, and thus enable us to apply latency optimization to the best-path selection performed afterward.

RTT Data. Using LARS (see § 3.1 for details), we generate 14.15Bn round-trip time samples in 84 measurement runs. Each run performs an ICMP-based RTT measurement to every IP in the address space advertised to the RS. At a low scanning rate of 100 Mbps, a single run completed in 3 hours. Our data collection took place between November 2025 and January 2026, over 54 days. Our latency data set is publicly available [18].

IP Geolocation Data. To geolocate IP addresses, we use the commercial geolocation dataset by IPInfo (see § 2.1), provided via a research partnership.

IXPs Traffic Weights. The IXP provided us with relative traffic weights at /24 prefix granularity to study if BGP routes actually carry traffic and to what extent. The traffic share is derived from the IXP's monitoring system and aggregated to the entire IXP, not on a per-member basis.

Limitation: IPv4 Focus. We observe that the IPv6 traffic share at the partner IXP is ≈ 9% and thus limit the evaluation in this section to IPv4. While traffic shares can vary between IXPs, this choice enables us to focus the evaluation on the currently dominant traffic class. We remark that LARS itself is not limited to IPv4. Extending the evaluation to IPv6 requires switching zmap to the IPv6-capable zmap6 [5], with the primary challenge being target address generation, since the IPv6 address space can no longer be fully enumerated. Thus, probing every IP in the address space advertised to the route server will no longer work, and the probing mechanism needs to switch to using a hitlist [26].

4.2 Ethical considerations

We carefully take a number of steps to ensure all data processed is in compliance with ethical standards. All measurements were performed in close coordination with the IXP.

Traffic Weights. The traffic weights are aggregated to /24 prefixes for the overall traffic. Hence, they do not enable to derive any traffic information on a per-member basis. Capturing the data is compliant with the local legal regulations and was performed as part of the IXPs monitoring system.

RTT Measurements. We follow the scanning best practices suggested by Durumeric et al. [19]. First, we set up a web server at the probes' source IP, serving a static page with explanatory information and a procedure to opt out of the scans. In addition, we set a reverse DNS entry to aid receivers in identifying the scan probes. Third, we use a low scan rate by limiting ZMap's available bandwidth to 100 Mbps.

Latency Comparisons. Our study involves latency measurements and traffic analysis, both can have ethical implications. While latency data is typically *not* considered sensitive, it is in our case as it can reflect the performance of the specific IXPs. Comparing such data across IXPs carries potential commercial implications, including impacts on the reputation of the IXPs involved. To address these concerns, we keep data confidential and only share aggregated results.

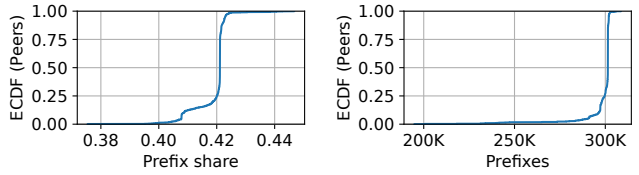
4.3 Presence of Route Diversity at the IXP

Route optimization in BGP best-path selection is possible only when there is route diversity. First, we investigate the presence of route diversity in the export tables of this specific IXP—similar to our study on IXPs at a global scale in § 2.2. The IXP's route servers are configured to hold export tables, i.e., all routes eligible for export to a peer based on the BGP communities of the advertisement. Consequently, we have ground truth on each peer's view of the route server after applying routing policies (action communities) and before best-path selection, and do not need to rely on reverse-engineered selective redistribution logic as in § 2.1. We exclude the monitoring AS at the IXP's RS, as it receives all routes.

Route-Diversity Exists. Figure 9a shows that every peer's export table contains diverse paths for at least 38% of the existing prefixes. For some peers, even up to 46% of the routes in the export table offer multiple paths to the same prefix. Figure 9b shows the absolute number of prefixes in the export tables at the RS. We find that almost every peer receives about 300k routes, which is ≈ 95% of the RIB size. As shown previously, it is expected that about 40%, i.e., ≈125k of them, have path diversity.

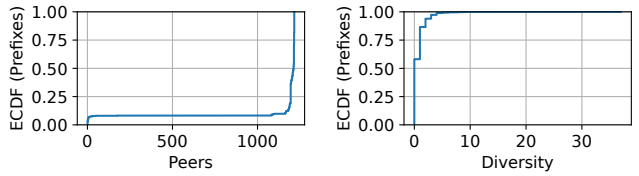
Prefixes with Path-Diversity are Widespread. When a prefix exhibits path diversity, that diversity is typically visible to many members. Figure 9c shows, for each diverse prefix, in how many peers' export tables it appears. For over 80% of peers, all prefixes with path diversity are present in their export table. In other words, if path diversity exists for a prefix, it is very likely to be available to all the peers at this specific IXP. This means that a large set of members can benefit from path optimizations by LARS.

There can be many diverse paths. Lastly, Figure 9d shows the degree of path diversity across all prefixes. We find that about 42% of the prefixes exhibit some degree of diversity, i.e., have alternate



(a) Prefixes with path diversity in the peer's export tables of the IXP RS.

(b) Total number of prefixes in the peer's export tables of the IXP RS.



(c) #peers each prefix with path diversity is sent to

(d) Degree of path diversity across all prefixes

Figure 9: Export table analysis of the IXP's RS.

paths. 28% of the prefixes in the RIB have exactly one alternate, and the most diverse prefix has 41 alternative paths. More available paths imply greater potential for latency optimization.

Takeaway. At this IXP, route servers expose substantial and broadly shared path diversity (≈38-46% of prefixes per peer, typically visible to almost all peers), often with at least one—and sometimes dozens (up to 41)—alternative paths, enabling RTT-based best-path optimization at scale.

4.4 Acquiring Path RTT Figures

Path RTT measurements. Over a period of 54 days, we retrieve RTT figures for responding hosts within *all paths* in the IXP's RS RIB. Having LARS configured to use a maximum bandwidth of 100 Mbps for outgoing traffic, LARS takes about 3 hours to scan the whole RIB of the IXP. In total, scanning all the paths in the RIB for a total of 84 runs resulted in over 17.97Bn responses, of which 14.15Bn are valid ICMP responses. We exclusively use these responses to derive RTT samples for a path.

Anomalous response patterns. During the scans, we observed a very small fraction of targets with unusual response patterns. Some targets responded to a single probe with an exceptionally high volume of ICMP messages. Overall, 0.00003% of scanned targets showed such behavior. We manually investigated the 10 most extreme cases using `ping` and `traceroute` and found targets that generated thousands of responses to a single ICMP echo request. Such an observation can be caused by misconfigurations at the targeted AS. We remove such duplicates in post-processing, ensuring that no target has more responses than paths.

Validation of transit path. To achieve more stable paths and higher hit rates, our measurement setup always measures the sum of the latencies of the forward path via the IXP to the host and the return path of the ICMP reply via our transit connection (see § 3.1). To avoid measuring a low-quality transit connection, we plot the path stability towards the measurement AS as seen from the 275

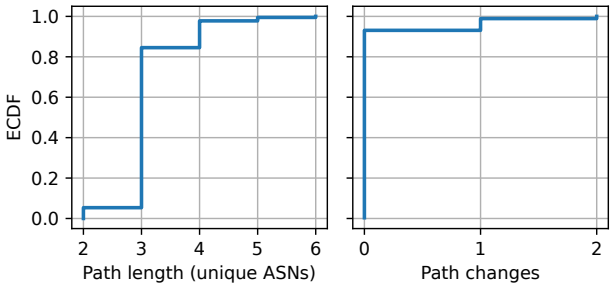


Figure 10: Our transit path is stable and via the transit, our measurement AS is close to all RIPE RIS ASes.

ASes providing collectors to the RIPE Routing Information Service (RIS) in Figure 10. First, we find that more than 80% of all collector ASes are no more than three hops away from our measurement AS, and 98% no more than four hops. Second, we find that there were hardly any changes in routing towards the transit prefix during our measurement period. More than 95% of the ASes did not see a single path change over the 54 measurement days, and none saw more than 2 path changes.

Takeaway. The LARS measurement setup provides a stable environment for RTT measurements that cover all paths in the RIB.

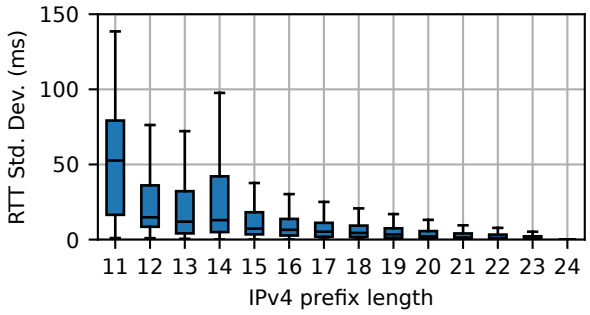
4.5 What's a good prefix length given RTTs?

A key design question for LARS is the granularity at which to attach RTTs to prefixes. This is rooted in the fact that, for large prefixes (e.g., /8) in internationally operating networks, RTTs can vary widely. Since operators typically want to avoid deaggregating the RIB (e.g., into /24s, the smallest routable unit in IPv4), we next study how often such large RTT spreads occur and whether they must be handled by LARS (e.g., by not tagging large prefixes with high RTT variance or by other means). This evaluation is possible because we measure every IP address in the RS address space.

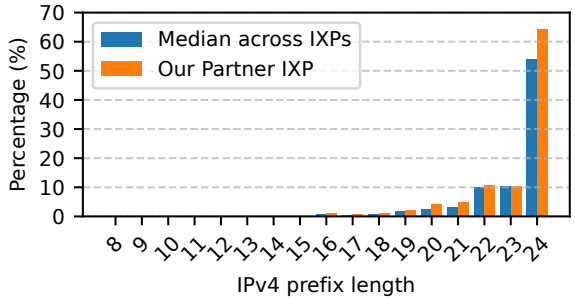
Large Prefixes Can Experience Latency Variability. Figure 11a shows that the standard deviation of RTT measurements declines with increasing prefix length. That is, there indeed exist large prefixes with notable variability in measured RTTs, as well as large prefixes with more consistent RTT figures. However, whether LARS needs to account for RTT variability depends on its likelihood.

Most Prefixes at IXPs are Long. To answer this, we show the prefix length distribution for *i)* all IXPs from § 2 and *ii)* the partner IXP in Figure 11b. The figure shows that the majority of prefixes at the studied IXPs are long, with /24 prefixes being the dominant case (>60% for our partner IXP). Recall that the previous evaluation showed that latency spread in a prefix is low for long prefixes that dominate the RS RIB. Thus, no special treatment for most prefixes is needed, and RTTs can be annotated directly in RIB entries.

Dismissed: de-aggregation of larger prefixes. The easiest way to solve the issue is to de-aggregate all larger prefixes actively before exporting them to member ASes. We discussed this proposal with the partnering IXP operator and some of the member ASes, and it was dismissed as too intrusive: while route aggregation helps minimize routing table size, de-aggregation would lead to routing table bloat. It would also make traffic engineering more complex,

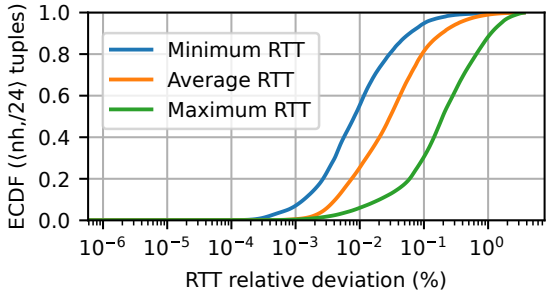


(a) RTT standard deviation per prefix size. Smaller prefixes exhibit higher variation.

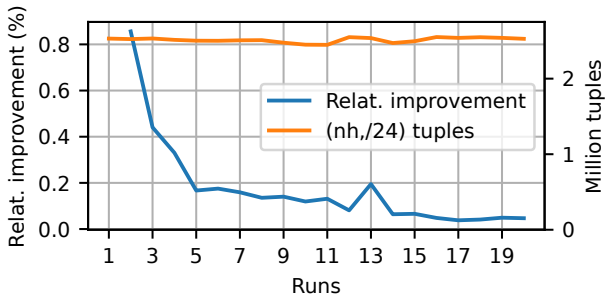


(b) Prefix length distribution at IXPs. Smaller prefixes are underrepresented.

Figure 11: Higher RTT consistency for more specific prefixes that are also the most frequent prefixes.



(a) Stability of estimates: distribution of relative deviation across multiple runs for each estimator.



(b) Estimator stability: relative improvement rate of the RTT estimates over runs (blue, left y-axis), and number of scanned (nh,/24) tuples per run (orange, right y-axis).

Figure 12: RTT Measurement stability & improvement

because a de-aggregated route could take multiple different AS paths, whereas there is just one without de-aggregation. Therefore, we dismiss the idea of de-aggregation.

Tagging and median. As de-aggregation is not an option, we tag larger prefixes with the median RTT of all covered /24s. If they exhibit high RTT spread, we add a special community tag. This community carries the semantics of a recommendation to de-aggregate the prefix on announcement, making the high spread visible to operators in the IXP's looking glass.

Takeaway. Most IXP routes consist of specific prefixes, predominantly /24, which exhibit low RTT spread and can be directly annotated. In contrast, the less common prefixes that show high RTT variability are also annotated but are tagged with a dedicated community label. This allows networks to filter these prefixes.

4.6 RTT Measurement Stability

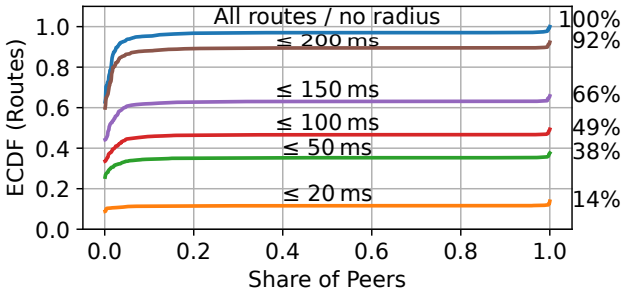
While measuring latency characteristics of paths across the Internet, our derived RTT samples can be affected by stochastic effects such as queuing or congestion. Next, we assess the impact of such effects.

Estimator stability. To avoid any larger geographic spread outlined in the previous section for the stability evaluation, we break the RIB into /24 prefixes by virtually de-aggregating all non-/24 prefixes. Using this data, we benchmark evaluators against each other to find the estimator with the highest stability over longer periods for each /24 and path combination, across the full 54-day measurement window.

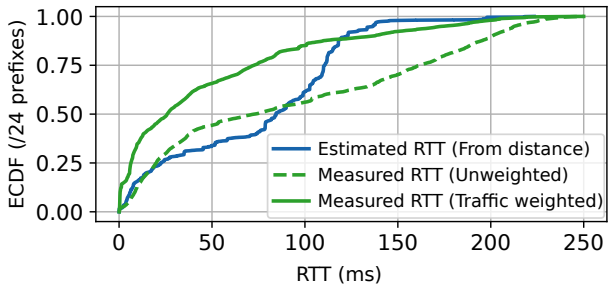
To do so, we calculate the minimum, mean, and maximum per measured /24 for each of our 84 measurement runs and plot an ECDF of their relative deviations across runs in Figure 12a. The result is that all three estimators are highly stable, with relative deviations within 1% across measurements in 90% of runs. The minimum estimator stands out as the most stable, with relative deviations below 1% for 99% of all measurements.

Selected RTT Estimator: Minimum Over /24s. Based on these findings, we chose the minimum estimator to annotate /24s with latency measurements. This also closely aligns with our goals for LARS: we want to provide stable, long-term estimates that are not influenced by transient effects such as congestion. Generally speaking, a packet's latency is the sum of its queuing, processing, transmission, and propagation delays. In the best case, i.e., empty queues, idle processors, and non-shared media, the latency approximates the propagation delay, which can be estimated using the min estimator with a sufficiently large number of samples. As the propagation delay is the only non-stochastic component of latency, it is a reasonably stable measurement target for LARS.

Required samples. Choosing the minimum estimator begs the question of how many measurements are actually needed to get sufficiently close to the true value of the propagation delay. Fig. 12b investigates how the relative improvement of our minimum-based RTT estimate decreases with any additional measurement runs on average per given /24 and path, i.e., how much additional measurement data can improve an estimate, and when stable estimates are achieved. Already from the second run onward, the average relative



(a) Share of routes redistributed by latency radius.



(b) Measured RTT and traffic-weighted RTT in comparison to our estimated RTT from the geolocated prefixes.

Figure 13: Latency radius evaluation

improvement per next-hop and /24 combination (nh,/24) drops below 1%; as more runs are added, the improvements quickly become smaller. That is, our 14-hour probing cycles yield stable RTTs (time not shown in the plot). To put the result into perspective, LARS can scan the IP space at our partner IXP's RS in roughly 3 hours, so running multiple scans per day is absolutely realistic.

Takeaway. RTT estimates are highly stable over time, with the minimum RTT per /24 being the most stable (≤ 1% deviation for 99% of measurements), so LARS uses it and, in practice, reaches stable estimates with few measurement runs (relative improvement < 1% after the first run, < 0.2% after the fifth run).

4.7 Latency Radius Evaluation

One of the most compelling features of LARS is the possibility of not receiving any (RTT-measurable) prefixes beyond a latency radius—neither directly from an IXP member nor from its customer cone. We discuss how this feature affects the IXP not only in the control plane but also in the data plane (i.e., the IXP switching fabric).

Share of redistributed routes. Figure 13a shows the share of routes received by share of peers for different (coarse-grained) latency bins. This plot accurately reflects selective redistribution and is normalized to the number of routes with measurable latency in the RIB. When limiting the latency radius to only the worst-offending routes with an RTT ≤ 200ms, the share of lost routes is as high as 8%; for a 150ms radius, it is 34%, and drops below 50% for a 100ms radius. Only 14% of the routes are within a radius of 20ms. This behavior is exactly what we want: by tightening the radius, an AS can deterministically prune high-latency options and thereby enforce tail-latency control, with the chosen threshold reflecting the operator's latency objectives (e.g., for anycast services such as root DNS or hyperscalers).

Traffic impact. As we received aggregated traffic data per /24 over a day from our partner IXP, we can quantify an upper bound on the traffic loss if all peers were to apply the same latency radius simultaneously. We can only derive an upper bound because not all traffic at IXPs is steered by RSes—traffic may also be steered by private peering sessions across the peering fabric. To do so, we plot three ECDFs in Figure 13b: the estimated RTT based on geolocation for our partner IXP (blue, similar to Figure 5 for the global view), the measured RTT by LARS (dashed green), and the traffic-weighted measured RTT by LARS (green). The ECDFs are calculated on a /24 basis, not a route basis, to handle differing prefix sizes.

The first observation is that the measured RTT (dashed green) and the estimated RTT (blue) follow comparable distributions up to 100ms, after which they deviate. The distance-based RTT estimate (blue) underestimates the share of long-tail routes at least for our partner IXP. While this observation cannot be generalized to other IXPs without more measurements, it is a hint that long-tail routes might even be more prevalent in the wild than reported in § 2.3. However, as observable from the traffic-weighted measured RTT (green), long-tail routes carry traffic less frequently than routes with shorter RTTs. The traffic-weighted measured RTT shows that a latency radius of 100ms applied by *all* peers would cost the IXP at most 10% of traffic. Whereas removing only long-tail routes above 200ms incurs almost no traffic loss and can solve headaches for member ASes in terms of traffic engineering.

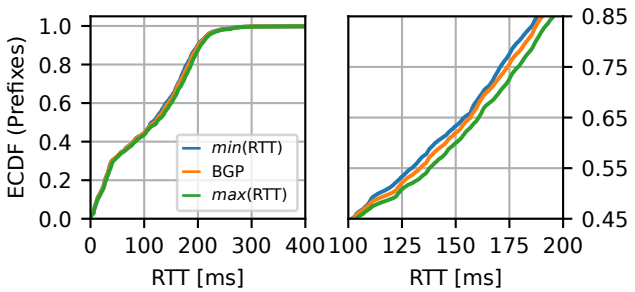
Takeaways. By design, LARS enables members to intentionally trade route-set size for stricter latency bounds: tightening the latency radius can substantially reduce the set of routes exported by the RS (e.g., by 86% for a 20ms constraint), while still allowing tail-latency mitigation above 200ms at almost no cost—minimal loss of traffic for the IXP and only minor route loss for the member.

4.8 Alternate Path Optimization Evaluation

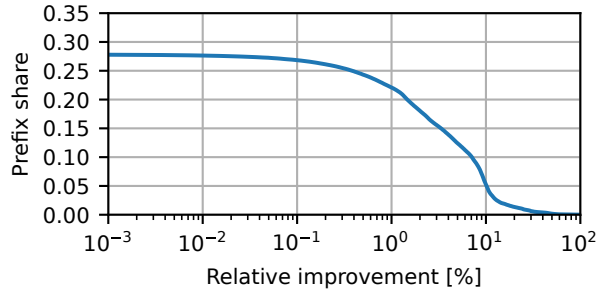
As an orthogonal mechanism to latency radii, LARS can also optimize according to RTT between alternate paths. We evaluate the benefits in the following.

LARS vs. BGP best path selection. Figure 14a shows the distribution of the minimum RTT and the maximum RTT towards all measured prefixes. The gap between the minimum and maximum RTT indicates the potential impact of selecting different alternate paths for the same prefix. We find this space to be surprisingly narrow, implying that BGP selects paths with acceptable latency for many prefixes—at least for our partner IXP. The most impact is found between 100ms and 200ms (see right in 14a).

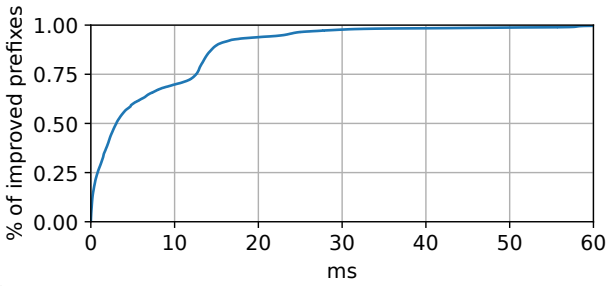
We present a different view of the same measurement data in Figure 14b. This CCDF (tail distribution) shows the potential RTT improvements *relative* to BGP's performance. We find that LARS is able to improve the RTT of about 28% of all prefixes present in the RS RIB, whereas for 5% of them, an RTT improvement by over 10% is achieved. While appearing modest at first glance, we plot the *absolute* RTT improvement for all improved prefixes in Figure 14c. We find that the majority see an absolute RTT improvement of over



(a) Empirical CDF of the prefixes RTT.



(b) CCDF of the potential RTT improvement relative to BGP.



(c) ECDF of the absolute RTT improvement for all improved routes.

Figure 14: Alternate path optimization evaluation.

4ms. For about 30% of them, however, their improvement goes well beyond 10ms, reaching up to 60ms in rare cases.

Takeaways. LARS can optimize between alternate paths and reach 10% or more RTT reduction for 5% of the prefixes, corresponding in some cases to up to 60ms RTT improvement. This is a notable gain in an already highly latency-optimized IXP environment where further latency improvements are non-trivial and valuable.

4.9 Route Server Overhead Evaluation

As we have access to the toolchain for generating RS configurations for a large IXP, we can evaluate LARS in a near-real-world environment using the same software stack used in production. This way, we evaluate the changes introduced by LARS to the RSes the same way our partner IXP does for new RS features.

Toolchain properties. The RS toolchain instruments multiple BIRD instances and maintains > 850 BGP sessions in total. Without LARS, the generated RS configurations have ≈5.5M lines of code, >97% of which are prefix filter sets for IRR (75.8%) and RPKI origin

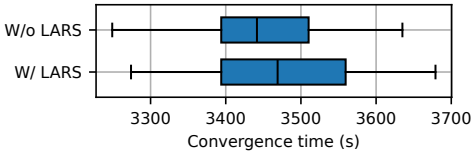


Figure 15: RS convergence time with and without LARS.

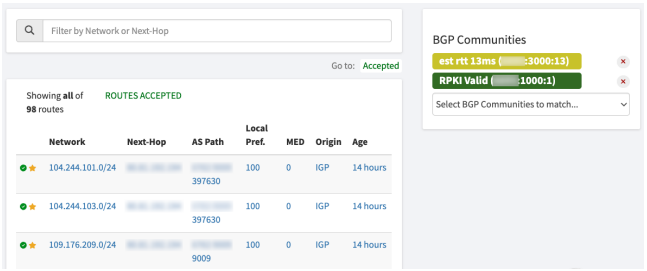


Figure 16: LARS LG integration with latency tags: filtering all RPKI valid routes with 13ms RTT for single AS.

validation[3] checks (21.7%). The toolchain runs six times per day at the partner IXP, generates configurations, and runs a soft reconfiguration on the running BIRD RS instances to load the changes.

RS configuration overhead. LARS adds roughly 400k lines of code to the existing configuration (7.16%) if we tag all routes with ≤ 100ms RTT with a 1ms resolution (i.e., 100 bins of 1ms). If we remove the embedded RPKI ROA checks from the configuration, LARS configuration overhead increases slightly to (8.16%). Compared to the share of IRR checks (70.39%), the overhead is fairly small. The added surplus lines consist mainly of the latency-tagging prefix and next-hop sets shown in Figure 8 and a small amount of logic.

Convergence evaluation. We simulate one RS instance and all peers in a container-based setup and measure the convergence time for all peers announcing their respective routes as extracted from a full RIB export of the partner IXP, i.e., the time from starting the experiment until no further changes happen to any RIB (peers or RS) anymore. This setup constitutes a stress test for the RS, resulting in roughly 1 hour of convergence time. A similar approach is used by the partner IXP to evaluate the overhead of RS features. It is therefore well suited to measuring the overhead of LARS.

We perform 30 convergence experiments for the RS setup, with and without LARS, using the aforementioned tagging granularity, and compare the convergence times. The measurements are presented in Fig. 15. The difference between the two setups is hardly measurable. When comparing median values, we find that LARS adds ≈ 30s to the time to full convergence.

Production-Level Prototype. We implemented LARS as a production-level prototype integrated into the partner IXP's operational route-server toolchain and LG software; see Fig. 16. The prototype validates passive route tagging in production: path-latency information is computed by LARS, attached to routes, and exposed in the LG. Active-mode policy enforcement has so far been evaluated through generated configurations, traffic-loss analysis, and convergence experiments rather than through an operational deployment.

[3]The partner IXP applies static RPKI origin validation without direct ingest via the RTR protocol; all RPKI data is present in the configuration.

To support a safe transition toward operational use, the prototype implements active-mode enforcement at *per-member granularity*: LARS can be enabled independently for individual member BGP sessions without changing the routes distributed to other members. This design enables incremental and non-disruptive deployment by the IXP operator. For example, members could opt in to LARS and configure their latency policy through a member portal.

Takeaways. LARS adds negligible overhead to the RS setup. The remaining deployment limitation is operational rather than computational: passive tagging is already validated for production, whereas active-mode policy enforcement should (and can) be rolled out incrementally on a per-member basis.

5 Related Work

BGP Performance. BGP path inflation is well studied, identifying the combination of BGP policies, Internet connectivity, cabling, and peering as its root cause [29, 30]. In the context of content provider networks, several works have studied opportunities for performance improvement. Schlinker et al. evaluated Facebook's user-perceived performance in terms of latency and goodput, and improvement opportunities through routing changes, showing minor to no opportunities [27]. Following on these findings, [10] searches for explanations why BGP, despite its shortcomings, is rarely far off from heavily engineered and expensive performance-aware routing schemes deployed by major content providers. Their results show that BGP's performance was comparable to that of performance-aware systems, leaving little room for improvement [10]. This paper shows that there is potential for latency optimization at IXPs that has, so far, gone unused.

Performance-Aware Routing. Performance-aware routing augments BGP's limited performance sensitivity by incorporating external measurements—e.g., delay, loss, or congestion—into egress-path selection. Such performance optimizations have, however, so far only been applied at edge networks—mostly hyperscalers. Facebook showed that leveraging performance metrics to steer traffic over alternate paths can reduce median latency, while also revealing the risk of oscillations in multi-path schemes [28]. Microsoft's FastRoute [20] reduces propagation and queuing delays via anycast-based ingress control, though such mechanisms rely on service nodes and are not directly applicable at IXPs. Performing ingress control at ISPs for performance optimization requires monitoring ingress traffic instead of using BGP [23]. Large operators have also deployed egress-control systems that combine routing visibility with performance telemetry. Google's Espresso [32] and Facebook's Edge Fabric [28] use BGP state, link-utilization data, and end-host performance to select higher-quality egress paths. Both platforms demonstrate that coordinated traffic shifting can improve utilization, avoid congestion, and reduce latency.

Research prototypes complement these industrial systems. Routescout [9] utilizes programmable-switch telemetry to estimate delay and loss for alternate BGP paths, achieving delay improvements in emulated settings while leaving questions of accuracy and scalability open. TANGO [13] enables coordinated smaller edge networks to expose additional BGP-compliant paths, gather fine-grained telemetry, and perform data-plane rerouting without relying on third-party infrastructure. Internet-wide experiments show that TANGO exposes paths that outperform default BGP for the majority of edge pairs, reducing latency by up to 39%. However, unlike LARS, TANGO requires AS-level coordination, whereas LARS works on existing route server infrastructure without coordination.

Latency and Location Communities at IXPs. Several IXPs already expose latency- or location-related route-server communities. IX.br documents RTT informational and action communities as well as location tags and location-scoped suppression; Equinix exposes RTT-threshold actions; and IXPs such as DE-CIX provide location-related tags or controls. These mechanisms show operational demand for latency and location awareness, but their latency signal is confined to the peering LAN or the member's router, i.e., it captures only the route-server-to-member port/router relationship. In contrast, LARS actively measures the RTT over the selected BGP path, spanning multiple AS hops, up to hosts within the announced prefix. This enables per-prefix suppression beyond a member-defined latency radius and latency-aware selection among alternative paths, including paths through a peer's whole customer cone.

BGP Mechanisms Beyond Communities. AIGP [24] can carry accumulated path-cost information in BGP, but targets controlled administrative settings and is non-transitive. It therefore cannot expose measured multi-hop latency from an IXP route server to arbitrary members. LARS fills this gap using standard communities.

Research Gap. Beyond a few efforts by large hyperscalers at the edge, no work has practically applied latency-based BGP optimizations in the core of the Internet—at IXPs. We close this gap by proposing LARS, which we show to be effectively deployable at a very large IXP.

6 Conclusion

This paper presents LARS, the first practical approach that enables IXP members to obtain latency-optimized routes via existing RSes. By evaluating LARS at a very large IXP, we show that this idea is no longer a vision; latency-aware routing can be deployed within today's RS toolchains without changes to BGP and without notable runtime overhead. The key enabler is that RSes expose substantial, broadly shared path diversity, making latency-based best-path selection feasible for many prefixes and members. LARS provides stable and well-covered RTT annotations by continuously probing the RS RIB, leveraging the minimum RTT per /24 as a robust long-term estimator. Operationally, members can intentionally trade route-set size for tighter latency bounds through a configurable latency radius, achieving tail-latency control at low cost, and can further reduce RTT by selecting among alternative paths—yielding measurable improvements even in an already latency-optimized IXP environment.

7 Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) grant XLerate (grant number 16KIS2055K and 16KIS2056). D. De Andres Hernandez's work is supported by ORIGAMI project (101139270) funded by SNS JU and the European Union.

References

[1] [n.d.]. . https://pulse.internetsociety.org/en/ixp-tracker/
[2] [n.d.]. . https://github.com/ds-ukassel/zmap-multipath-rtt
[3] [n.d.]. . https://bird.network.cz/
[4] [n.d.]. . https://www.ixpmanager.org/
[5] [n.d.]. . https://github.com/tumi8/zmap
[6] [n.d.]. *Overview of MegaIX Features.* https://docs.megaport.com/ix/overview-features/
[7] 2019. *IX.br BGP Communities Policy.* https://old.ix.br/doc/politica-de-tratamento-de-communities-no-ix-br-v3_0-english.pdf
[8] Yasin Alhamwy, Bashar Khoulani, and Oliver Hohlfeld. 2025. Crawling Alice Looking Glasses at IXPs to Quantify BGP Route Diversity. In *Proceedings of the ACM SIGCOMM 2025 Posters and Demos*. 121–123.
[9] Maria Apostolaki, Ankit Singla, and Laurent Vanbever. 2021. Performance-Driven Internet Path Selection. In *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*.
[10] Todd Arnold, Matt Calder, Italo Cunha, Arpit Gupta, Harsha V. Madhyastha, Michael Schapira, and Ethan Katz-Bassett. 2019. Beating BGP is Harder than We Thought. In *ACM HotNets*.
[11] Robert Beverly. 2016. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *ACM IMC*.
[12] Debopam Bhattacherjee, Waqar Aqeel, Sangeetha Abdu Jyothi, Ilker Nadi Bozkurt, William Sentosa, Muhammad Tirmazi, Anthony Aguirre, Balakrishnan Chandrasekaran, P Brighten Godfrey, Gregory Laughlin, et al. 2022. cISP: A Speed-of-Light Internet Service Provider. In *USENIX Networked Systems Design and Implementation*.
[13] Henry Birge-Lee, Sophia Yoo, Benjamin Herber, Jennifer Rexford, and Maria Apostolaki. 2024. TANGO: Secure Collaborative Route Control across the Public Internet. In *USENIX Networked Systems Design and Implementation*.
[14] Ilker Nadi Bozkurt, Waqar Aqeel, Debopam Bhattacherjee, Balakrishnan Chandrasekaran, Philip Brighten Godfrey, Gregory Laughlin, Bruce M Maggs, and Ankit Singla. 2018. Dissecting latency in the Internet's fiber infrastructure. *arXiv preprint arXiv:1811.10737* (2018).
[15] Ignacio Castro, Juan Camilo Cardona, Sergey Gorinsky, and Pierre Francois. 2014. Remote peering: More peering without internet flattening. In *ACM CoNEXT*.
[16] Nikolaos Chatzis, Georgios Smaragdakis, Anja Feldmann, and Walter Willinger. 2013. There is more to IXPs than meets the eye. *SIGCOMM Comput. Commun. Rev.* (2013).
[17] Omar Darwich, Hugo Rimlinger, Milo Dreyfus, Matthieu Gouel, and Kevin Vermeulen. 2023. Replication: Towards a Publicly Available Internet Scale IP Geolocation Dataset. In *ACM IMC*.
[18] David de Andres Hernandez, Yasin Alhamwy, Daniel Wagner, Maximilian Stephan, Matthias Wichtlhuber, and Oliver Hohlfeld. 2026. *Dataset from: LARS: Keeping Local Traffic Local with Latency-Aware Route Servers at IXPs*. https://doi.org/10.5281/zenodo.21026109
[19] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications.. In *USENIX Security*.
[20] Ashley Flavel, Pradeepkumar Mani, David Maltz, Nick Holt, Jie Liu, Yingying Chen, and Oleg Surmachev. 2015. FastRoute: A scalable load-aware anycast routing architecture for modern CDNs. In *USENIX Networked Systems Design and Implementation*.
[21] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 681–692.
[22] Fabricio Mazzola, Pedro Marcos, Ignacio Castro, Matthew Luckie, and Marinho Barcellos. 2022. On the Latency Impact of Remote Peering. In *Passive and Active Measurement Conference*.
[23] Stefan Mehner, Helge Reelfs, Ingmar Poese, and Oliver Hohlfeld. 2024. IPD: Detecting Traffic Ingress Points at ISPs. In *ACM SIGCOMM*.
[24] P. Mohapatra, R. Fernando, E. Rosen, and J. Uttaro. 2014. *RFC 7311: The Accumulated IGP Metric Attribute for BGP.* Internet-Draft rfc7311. Internet Engineering Task Force. Standard.
[25] Ryo Nakamura, Kazuki Shimizu, Teppei Kamata, and Cristel Pelsser. 2022. A First Measurement with BGP Egress Peer Engineering. In *Passive and Active Measurement*, Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser (Eds.). Springer International Publishing, Cham, 199–215.
[26] Erik Rye and Dave Levin. 2023. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *ACM SIGCOMM*.
[27] Brandon Schlinker, Italo Cunha, Yi-Ching Chiu, Srikanth Sundaresan, and Ethan Katz-Bassett. 2019. Internet Performance from Facebook's Edge. In *ACM IMC*.
[28] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. 2017. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *ACM SIGCOMM*.
[29] Neil Spring, Ratul Mahajan, and Thomas Anderson. 2003. The Causes of Path Inflation. In *ACM SIGCOMM*.
[30] Hongsuda Tangmunarunkit, Ramesh Govindan, and Scott Shenker. 2001. Internet path inflation due to policy routing. In *Scalability and Traffic Control in IP Networks*. International Society for Optics and Photonics, SPIE.
[31] D. Walton, A. Retana, E. Chen, and J. Scudder. 2016. Advertisement of Multiple Paths in BGP. https://www.rfc-editor.org/rfc/rfc7911.
[32] KK Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, TaeEun Kim, Ashok Narayanan, Ankur Jain, Victor Lin, Colin Rice, Brian Rogan, Arjun Singh, Bert Tanaka, Manish Verma, Puneet Sood, Mukarram Tariq, Matt Tierney, Dzevad Trumic, Vytautas Valancius, Calvin Ying, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *ACM SIGCOMM*.

Appendices are supporting material that has not been peer-reviewed.

A Multipath RTT Probe Modules

Following, we detail how the mechanisms used by the contributed probe modules [2] work. They all leverage that responsive addresses will answer with a packet whose header will include the encoded information:

- *ICMP echo request messages* trigger echo responses, returning the request payload (available in the default `ZMap`). To enable multipath probing, we encode the sending timestamp and source MAC address into the payload and decode it from the response to generate the sample. Additionally, the first and second validation words generated by `ZMap` are stored in the identifier and sequence number fields, respectively. Figure 17 (upper part) illustrates this encoding having the added data fields depicted in cyan.
- *UDP* probes trigger useful responses for *unused* ports by generating ICMP port unreachable responses, which include the request UDP header. We thus encode the sending timestamp and egress interface into the length and checksum fields with an appropriate payload that makes their values legitimate, as in [11]. We do not show the header in Figure 17, as it is trivial.
- For *TCP*, we use the handshake and hand-off procedures to trigger responses. *TCP-SYN packets* should be answered with TCP-ACK packets; thus, the SYN number can be retrieved as $\text{SYN}_{number} = \text{ACK}_{number} - 1$. We encode the timestamp and MAC index in the sequence number field (cyan), as illustrated in Figure 17. In the case of *TCP-ACK packets*, the responses are TCP-RST packets echoing the ACK number. Here, the timestamp and MAC index is encoded in the acknowledgment field (orange). Like UDP, TCP probes arriving at unused ports will trigger ICMP port unreachable messages—we also use them to generate samples.

A.1 Response Validation Details

In its standard configuration, `ZMap` issues a single probe, so any duplicate response can be unambiguously interpreted as anomalous and safely discarded. Our methodology explicitly enables multipath measurements by sending multiple probes to the same target, one per candidate path. In this setting, `ZMap` validates the first received response and flags all subsequent ones as duplicates. As a result, when a target is reachable over k distinct paths and correctly returns k responses—one per probe, $k-1$ responses are classified as duplicates. While this behavior is correct, it renders duplicate signaling uninformative for distinguishing between legitimate multipath responses and genuinely anomalous ones. We remove such duplicates in post-processing, ensuring that no target has more responses than paths.

B Selecting a Probe Module

Since probes can be discriminated by protocol, the probe choice can yield different RTT views. To study the implications of this effect, we survey a random set of 1M hosts, once for each probe module. We restrict the comparison to the subset of targets who responded to all four probe types. The very similar distributions, shown in Figure 19, indicate that all four modules offer comparable RTT views. We also conclude that protocol overloading does not affect RTT, as the TCP and UDP modules provide a view similar to that of the non-overloaded ICMP module.

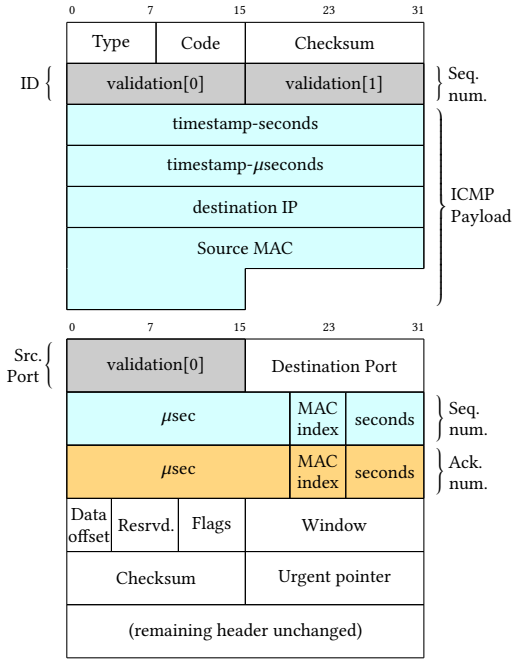


Figure 17: Usage specification of the ICMP fields in the `ZMap` ICMP multipath module (upper), and overloaded fields in the TCP modules (lower).

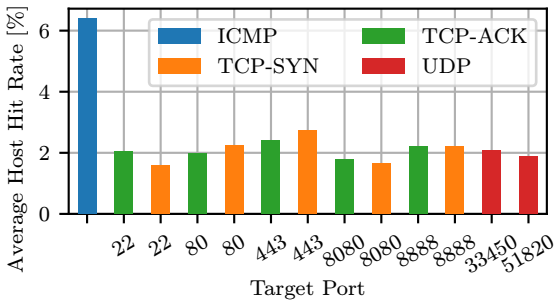


Figure 18: Hit rate by probe module and target port.

Having all four probe types offer equivalent latency views, we reduce our selection criterion to the one achieving the highest hit rate, i.e., the most latency samples, which is desirable. Without prior work comparing hit rates across different latency-probing methods, our evaluation can inform follow-up work. We thus compare the average hit rate for each probe module over the same set of 1M hosts. For the UDP/TCP modules, we also varied the target port across the five most popular at our partner IXP. As shown in Figure 18, we observe that ICMP achieves a 6% hit rate, while the TCP and UDP modules all stay below 3%. Given these results and ICMP's comparative simplicity, we find the ICMP module the fittest for generating round-trip propagation delay estimations. Thus, we

Table 1: The IXPs we scrape. The numbers refer to how many of the IXP's RSes fall into that category.

IXP	Small	Medium	Large
LINX	28	7	3
SMFIX	1	3	0
1-IX EU	1	6	0
1-IX Net	2	2	0
AMS-IX	50	0	0
BBIX	0	2	0
BCIX	0	4	0
DartPoints Bridge IX Dublin	8	0	0
CIX	2	0	0
DATAIX	12	10	2
DD-IX	4	0	0
DE-CIX India	12	4	0
DE-CIX	136	32	4
EVIX	4	0	0
FIX-Berlin	2	0	0
GNM-IX	7	11	2
IIX	3	0	2
INX-ZA	5	3	0
IRAQ-IXP	4	0	0
IX Australia	6	8	0
IX.br	120	24	11
IX-Denver	0	2	0
IX-KW	4	0	0
NZIX	4	2	0
LONAP	0	2	0
Megaport	179	9	0
MyIX	0	2	0
Netnod	19	4	0
nine-ix	2	0	0
Stuttgart-IX	4	0	0
Sibir-IX	2	0	0
SON-IX	2	0	0
TOP-IX	0	2	0
TURKIX	4	0	0

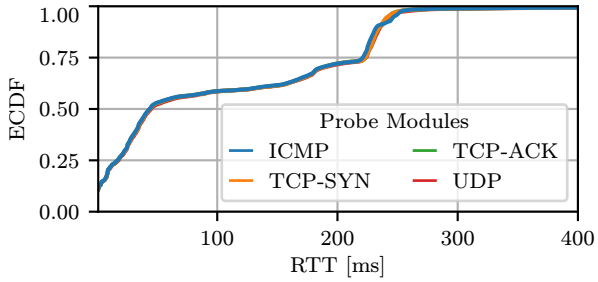


Figure 19: ECDF of the round-trip propagation delay of responsive networks from a 1M-targets set. All four modules produce equivalent results.

conducted the remaining experiments solely using the ICMP probe module.

C Alice LG Table

Table 1 shows the IXPs for which we scrape LG data using Alice LG and the number of RSes that fall under each category: small (less than 100 peers), medium (100-500 peers) and large (500+ peers). The AMS-IX LG sits on top of an external collector, not on top of the AMS-IX RSes; therefore, its numbers are not indicative of its size.