HiPerConTracer 3.0: Transport-level Packet Routing Analysis Tool

Thomas Dreibholz
Center for Resilient Networks and Applications
Simula Metropolitan, Norway
dreibh@simula.no

Somnath Mazumdar
Department of Digitalization
Copenhagen Business School, Denmark
sma.digi@cbs.dk

***Abstract*—Network-based applications rely on the underlying network infrastructure to reliably forward packets between nodes. The way packets are forwarded has a significant impact on service quality. Therefore, it is important to gain a better understanding of data packet routes. To obtain detailed information about network paths, continuous and long-term packet analysis is required. To achieve this, we present our open-source framework HiPerConTracer 3.0 for large-scale IP trace analysis. It performs Ping and Traceroute measurements to provide detailed insights into packet routes and packet timing by tracing routes between senders and receivers in public and private networks. Particularly, it runs its own measurements, without need to obtain data, or cooperation from, the underlying network service providers or remote server owners. Our tool supports large-scale data collection, storage, and post-processing stages. It supports easy-to-understand route visualization, round-trip time measurements, and hop counts. A proof-of-concept analysis revealed that packet route lengths can change drastically when traveling through unexpected countries, regions, and network operators.**

***Index Terms*—Analysis, Network, Packets, Routing, Round-Trip Time, Traceroute**

I. Introduction

The Internet Protocol (IP) provides connectionless, unreliable, and best-effort transmission of data packets (for both IPv4 and IPv6) between nodes. The IP only considers routing packets if possible; thus, it does not provide packet delivery guarantees. Packets may be delayed or dropped in the event of congestion or errors. Therefore, protocols on top of IP, (i.e. usually transport protocols like the Transmission Control Protocol (TCP) or the Stream Control Transmission Protocol (SCTP) [1]) have to cope with these properties to provide features like flow control, congestion control, and reliable transmission. Alternatively, the simpler User Datagram Protocol (UDP) just lets the application handle the performance constraints of the network.

In this context, two of the most popular tools to measure latency and obtain network paths from a network user's perspective are `ping` and `traceroute`. `Ping` works by sending an Internet Control Message Protocol (ICMP) [2], [3] `Echo Request` over IP from a source to a remote destination. The destination then responds with an ICMP `Echo Reply`, and the time between sending the request and receiving the response is known as the round-trip time (RTT). `Traceroute` uses the Time-to-Live (TTL) field in the IPv4 header [4] or the Hop Limit field in the IPv6 header [5], which is decremented each time a packet is routed. If the TTL/Hop Limit reaches zero before the packet arrives at its destination, the packet is dropped, and an ICMP `Time Exceeded` [2], [3] error message is sent back to the sender with the source IP address set to the router's address. By probing with increasing TTL/Hop Limit, the sender can detect the sequence of routers (`Time Exceeded` messages) until the destination is reached (`Echo Reply` message). In the case of a missing reply, the router is unknown. However, TTL/Hop Limit gaps in the responses reveal the number of unknown routers and their positions in the router path sequence. Overall, `ping` and `traceroute` are simple shell tools. A more feature-rich tool is required for accurate, long-term, high-frequency measurements for transport protocols from various endpoints in both public and private networks.

Here, we aimed to answer our research question, i.e., *How much delay can transport protocols anticipate from IP in today's Internet infrastructure?* For better analysis, the research question is further divided into three sub-research questions: *i)* Which routes do the packets take? *ii)* How much is the latency? and *iii)* How frequently do packet route changes occur in both long and short term? To answer these sub-research questions, we developed the transport-level packet routing analysis tool HiPerConTracer 3.0[1], which provides two primary capabilities: *i)* route tracing to obtain information about the intermediate routers; and *ii)* round-trip time measurements with high accuracy.

The scope of this paper is to present HiPerConTracer 3.0 as a tool to perform measurements in public and in private networks. It is not intended to find novel network effects or describe the same old network effects which are already mentioned in many scholarly works. Our proof-of-concept experiment aims to demonstrate the capabilities of HiPerConTracer 3.0 by performing a networking analysis to some well-known Internet servers, and to show that some (expected) properties of the network can be detected and shown solely based on HiPerConTracer 3.0 measurements, without the need for cooperation or support by the providers of servers or the underlying networks.

[1] HiPerConTracer 3.0 "TARTAN" is the experimental version based on HiPerConTracer 2.0.

Table I: HiPerConTracer and its evolution

Version	Features	Year	Ref.
HiPerConTracer 1.0	Measuring Ping and Traceroute between multi-homed systems.	2020	[10]
HiPerConTracer 2.0	UDP module, high-precision time-stamping	2024	[11]
HiPerConTracer 3.0	TCP module, jitter measurement	2025	–

II. Related Work

In the literature, there are many tracing tools which use their own trace data, BGP data and Ping/Traceroute data. In this section, we only consider works that have been published in the last five years. Maier et al. investigated persistent routing loops in the Internet, taking into account both IPv4 and IPv6 [6]. Luttringer et al. used Traceroute and Ping probes to detect all Multi-Protocol Label Switching (MPLS) tunnels along a route [7]. McCherry et al. sought to create multi-layer maps of the Internet infrastructure by geo-locating routers between two endpoints [8]. Huang et al. proposed a tool for rapid Traceroute in IPv4 address spaces only [9]. This work focuses on speeding up Traceroute scans of the IPv4 space, while we offer precise RTT measurements, and hop count for both IPv4 and IPv6.

III. Overview of HiPerConTracer 3.0

Here, we are interested in knowing how data packets flow from source to destination without presumptions and without relying on ISPs' support. Gaining insight into latency-related issues in an application requires accurate measurements of latency and an overview of the underlying routing. To do this, Ping and Traceroute measurements should be performed regularly and with sufficiently high frequency from appropriate vantage points.

Existing infrastructures – such as Archipelago[2] (Ark), RIPE Atlas[3], and NorNet[4] – provide Ping and Traceroute measurements from fixed vantage points on the Internet. However, due to their location on the *public* Internet, they cannot provide detailed performance related insights into private networks (i.e. extranets and intranets). Monitoring such systems requires *own* vantage points, particularly also allowing to perform long-term, high-frequency measurements with customized settings. Existing command-line tools like `ping` and `traceroute` are not suitable for this purpose, since these tools just produce simple, human-readable output for simple source/destination runs. Furthermore, a suitable data storage and retrieval system is also needed for long-term analysis. To address this need, we developed our open source tool called High-Performance Connectivity Tracer[5] (HiPerConTracer).

A. Key Features

HiPerConTracer 3.0 performs large-scale, customized Ping and Traceroute measurements. We avoided BGP routing data, because the BGP data are not primarily intended to be used to infer autonomous system (AS)-level mappings. By definition, BGP was not designed with an AS-level topology discovery feature [12]. Also, we do not want to rely on information provided by ISPs, since it is unlikely to obtain such data from ISPs in most cases anyway, due to security and business reasons. Furthermore, the RIPE Atlas infrastructure, which provides Traceroute measurements, does not always accurately depict global Internet connectivity [13]. HiPerConTracer 3.0 provides a wide range of advanced features, including:

- multi-transport-protocol support (ICMP, UDP, TCP);
- multi-homing and parallelism support;
- handling of load balancing in the network;
- multi-platform support (currently Linux and FreeBSD);
- high-precision (nanoseconds) timing support [11] (Linux timestamping, both software and hardware);
- a library (shared and static) to integrate measurement functionality into other software (libhipercontracer);
- database import ("Importer Tool") for SQL (MariaDB/MySQL, PostgreSQL) and NoSQL (MongoDB);
- database export ("Query Tool") with filter to e.g. query certain measurements and/or time ranges;
- data processing tool ("Results Tool") to e.g. convert data to comma-separated value (CSV) file format;
- open source and written in a performance- and portability-focused programming language (C++).

In the following, we are going to describe the key features in more detail.

Table I shows the evolution of HiPerConTracer. While the programs basic Ping/Traceroute measurements are similar in all versions (i.e. their performance is the same), the difference is in the support of features like high-precision timing as well as additional transport protocols.

1) Multi-Transport-Protocol Support: Ping and Traceroute traditionally use ICMP, which relies on the `Echo Request`/`Echo Reply` mechanism of the ICMP protocol [2], [3]. ICMP is provided by all IP devices. Therefore, it does not require the remote device to have any special services running. The `Echo Reply` is generated directly by the network stack in the kernel. This also means that it does not need a context switch to user space. However, it is possible that the kernel settings or firewalls may block this type of probing traffic.

Therefore, an alternative is to use UDP instead. However, this requires that the remote system has a UDP Echo service [14] installed. The UDP Echo services simply sends ("echoes") incoming UDP packets back to the sender. A UDP Echo service is typically provided by a small user space program, i.e. a context switch to user space is required, adding a small bit of additional latency. Alternatively, a router/firewall can be configured to act as "reflector", echoing the UDP packets in hardware. Together with HiPerConTracer's high-precision timing support, this allows for very accurate timing measurements [11], [15]. In any case, i.e. using UDP Echo service or a packet reflector configuration, applying UDP for Ping or Traceroute measurements requires support by the

[2] Archipelago (Ark): https://www.caida.org/projects/ark/.

[3] RIPE Atlas: https://atlas.ripe.net/.

[4] NorNet: https://www.nntb.no/.

[5] HiPerConTracer: https://www.nntb.no/~dreibh/hipercontracer/.

remote system. If this support is unavailable, i.e. the usual case for public servers, UDP cannot be used.

However, public servers necessarily provide some kind of publicly accessible service, typically via TCP on a certain TCP port, e.g. port 80 for the HyperText Transfer Protocol (HTTP) or port 443 for HTTP Secure (HTTPS) on web servers. As public servers, they certainly must answer a TCP SYN packet with a TCP SYN+ACK packet, as part of the three-way TCP handshake [16]. A feature added by HiPerConTracer 3.0 is therefore Ping/Traceroute using TCP SYNs packets. Although routers handle these packets in the same way as ICMP and UDP when the TTL/Hop Limit reaches zero (i.e. an ICMP `Time Exceeded` is sent back to the sender), the end-system finally answers with a TCP SYN+ACK packet. In this case, the local instance finally generates a TCP RST packet, informing the remote system to immediately release the connection resources again. This prevents affecting the remote system unnecessarily by the measurements, and avoids misinterpreting the measurements as a *SYN-flooding* attack [17].

2) Multi-Homing and Parallelism: The aim is to particularly support multi-homing, which includes dual-homed IPv4/IPv6 systems, routers (connected to multiple networks), as well as systems connected to multiple Internet service providers for redundancy. By allowing Ping/Traceroute measurements to be conducted in different networks simultaneously, it is possible to take into account the independence of different networks.

3) Load Balancing: Routers may perform load balancing when multiple routes have equal costs. If workload balancing is only based on the source/destination address pair, fixed addresses ensure that all packets between these addresses take the same route. However, although TCP is based on the connectionless service of IP, TCP-like loss-based congestion control [18] makes the implicit assumption that (almost all) packets of a connection take the same route. The load balancing of packets of the same connection would cause packet reordering; gaps in the arrival sequence would then be misinterpreted as loss caused by congestion. Then, the result would be a very poor performance. Therefore, load balancing typically also takes into account the first four bytes of the Transport Layer header (e.g. TCP [16] or UDP [19] header), which usually contains the source and destination ports. This ensures that all packets of the same flow take the same route, while packets of different flows may take different routes. HiPerConTracer 3.0 therefore ensures that the source and destination ports for TCP and UDP, in addition to the source and destination IP addresses, are kept constant, to prevent load balancing distorting the measurements.

However, ICMP does not use port numbers [2], [3]. The only modifiable field in the first four bytes of its header is the ICMP checksum. Therefore, while for TCP and UDP the port numbers must be fixed to avoid load balancing via different routes, an `Echo Request` is specifically crafted by HiPerConTracer to keep the ICMP checksum constant. In addition, all measurement packets are sent in a burst. The required length of a Traceroute burst (i.e., all TTLs/Hop Limits from one to the necessary number to reach the destination) is stored after the initial block-wise probing. This burst further minimizes the probability of distorting a measurement run due to load balancing.

4) High-Precision Timing Support: RTT can be calculated by asking the system for the time before sending a request and then comparing it to the time after the response is received. However, this process is not very accurate when performed in user space, because of context switching. Linux offers the `SO_TIMESTAMPINGNS` feature providing software and hardware timestamps, if the network interface card and its driver support it. This feature can be used by HiPerConTracer to measure the RTT with nanoseconds precision [11]. It allows to detect fine-granular performance changes for network and server tuning [11]. In addition, for measurements to destinations under own control, the remote side (reflector) could send a response in hardware [15], allowing very accurate measurements of packet RTTs within the network without distortion by end-systems.

5) Database Import and Export: The efficient storage and retrieval of measurement results collected from various vantage points necessitates proper data management. It is essential to have the ability to import to and export from database management systems (DBMS) such as SQL or NoSQL databases. The actual choice of a DBMS depends on a user's needs, e.g. to use a familiar DBMS, use of an existing setup, or certain performance criteria for analysis queries. Currently, HiPerConTracer supports MariaDB/MySQL, PostgreSQL, and MongoDB. However, depending on the amount of data and the application scenario, a different DBMS system may be more suitable. Therefore, it is also important to have an extensible import/export interface for a database, where tools are provided to conveniently convert results into a suitable format for analysis, such as comma-separated values (CSV) files. For this purpose, HiPerConTracer provides Importer Tool, Query Tool, and Results Tool.

B. Data Format

HiPerConTracer 3.0 works with the data shown in the Listing 1. The output is in the form of plain-text tables, but the tools also have the capability to compress and decompress on-the-fly in GZip, BZip2 and XZ formats to save storage space.

The Sublisting 1a presents a Ping example, consisting of:

- Type (#Pi for ICMP Ping; #Pu for UDP; #Pt for TCP);
- Measurement ID (e.g. vantage point number, here: `0`);
- Source and destination IP address (here: IPv4);
- Time stamp (nanoseconds since January 1, 1970, 00:00:00 UTC) in hexadecimal;
- Burst sequence (here: `0`);
- IP traffic class (hexadecimal, here: `0x00`);
- Request and reply sizes (here: 64 B of `Echo Request`/`Echo Reply` including IP header);
- ICMP checksum (hexadecimal);
- Source and destination port (here: `0`, since ICMP is used);

(a) An Example of Ping using ICMP over IPv4

```
#Pi 0 10.44.33.111 193.99.144.80 1790bfc57c0d5753 0 0 44 44 dfd9 0 0 255 116666aa 42845 12379 97454 22516664 22363986 22284000
#Pi 0 10.44.33.111 193.99.144.80 1790bfc5c36b71b6 0 0 44 44 dfd8 0 0 255 116666aa 47264 25789 130818 22477446 22273575 22199468
#Pi ...
```

(b) An Example of Traceroute using TCP over IPv6

```
#Tt 0 2001:700:712:52:baca:3aff:fe92:9517 2a02:2e0:3fe:1001:302:: 1790bfc57c4f95e4 0 22 0 64 0 55027 80 200 7beb80468a5236bb
	1790bfc57c4f95e4 1 112 1 116666aa 24044 8047 133547 30816149 30650511 30549063 2001:700:712:52::1
	1790bfc57c4ec48b 2 112 1 116666aa 25245 7207 98295 9820183 9689436 9560844 2001:700:712:ff00::2
	...
	1790bfc5802ff861 22 64 255 116666aa 16506 6604 58848 22999863 22917905 22773063 2a02:2e0:3fe:1001:302::
#Tt ...
```

Listing 1: Sample representation of HiPerConTracer 3.0 Ping and Traceroute

- Status (e.g. 255 = response received, 1 = `Time Exceeded`);
- Timing information (i.e. how the following timing information was obtained to assess accuracy [11]);
- Send delay from application to queuing in kernel (in nanoseconds);
- Scheduling delay in kernel (in nanoseconds);
- Receive delay from kernel to application (in nanoseconds);
- Application RTT (in nanoseconds, from system clock in user space);
- Software RTT from kernel (in nanoseconds);
- Hardware RTT from network interface card (in nanoseconds).

The Traceroute format, which is demonstrated in Sublisting 1b, is quite similar. The first line provides an end-to-end description, while the subsequent tabulator-indented lines contain information about each hop (in this case, 1…22). This information includes the time taken to reach the routers and their IP addresses, with the last hop being the destination system. The IP address (here: IPv6) of each hop is provided in the last column.

C. System Architecture

Figure 1 shows the architecture of HiPerConTracer 3.0, where nodes that provide a view of the system run the measurements. The results are initially stored locally. The importer (Importer Tool) can run on a vantage point node, but in many cases it is better to first transfer the data to a dedicated importer server (Sync Tool) and perform the import there. This keeps the amount of storage space needed on the nodes small, while providing enough storage space to handle database outages (e.g. for maintenance) on the importer server. The database is a DBMS server (or cluster) running the chosen DBMS (such as MariaDB/MySQL, PostgreSQL, or MongoDB). In particular, data are stored in suitable schemata (i.e. SQL tables with appropriately typed columns for the recorded values, or NoSQL JSON objects for each measurement result), allowing for advanced filtering and joining of data using the features of the underlying DBMS system. Depending on the application requirements, indices and partitions can be configured to improve query performance. Analysis systems, e.g. a user's statistics software like GNU R, or a performance monitoring system, can then query the database for the desired results. Alternatively, the results can be exported by the Query Tool to the results files (in their original format), or be converted into CSV format by the Results Tool. This gives users full flexibility to choose their analysis tools.

IV. Experimentation

The advantages of HiPerConTracer 3.0 can also be utilized in one's own setup, where fixed measurements of public infrastructures such as RIPE Atlas are either inadequate or not feasible.

A. Experimental Setup

A measurement setup was performed on a Dell Precision T3600 with an Intel 82579LM Gigabit Ethernet interface, hosted at SimulaMet in Oslo, 🇳🇴 Norway. This interface and its driver support hardware and software timestamping capabilities[6]. HiPerConTracer 3.0 Ping and Traceroute measurements were performed from August to October 2023, to all public Comprehensive TeX Archive Network (CTAN) and Comprehensive R Archive Network (CRAN) web servers via TCP. That is, the experiment demonstrates HiPerConTracer's capabilities to run frequent, high-precision Ping/Traceroute measurements to various destinations. Please note that TCP is used here, since ICMP is firewalled at most destinations, and UDP would even require the server operators to deploy a UDP Echo [14] service.

The analysis was performed using GNU R, with IP address geo-location based on HLOC and IPinfo.io. HLOC is a measurement tool that utilizes the RIPE Atlas infrastructure to estimate IP address geo-locations by measurements from known vantage points on the Internet [20]. We used the HLOC results when their estimated distance was $\leq$25 km. For addresses without a useful HLOC location, we instead made online queries to the IPinfo.io[7] service. We used AS information from the CIDR Report[8] and AS number lookup from the free GeoLite2[9] database. That is, the mapping of IP addresses to ASs is based on the data provided by the Regional Internet Registries (RIR).

[6] For this wide-area network measurement, the high accuracy is not really necessary. Nevertheless, HiPerConTracer and our setup provides it.

[7] IPinfo.io: https://ipinfo.io.

[8] CIDR Report AS list: https://www.cidr-report.org/as2.0/autnums.html.

[9] GeoLite2: https://dev.maxmind.com/geoip/geoip2/geolite2/.

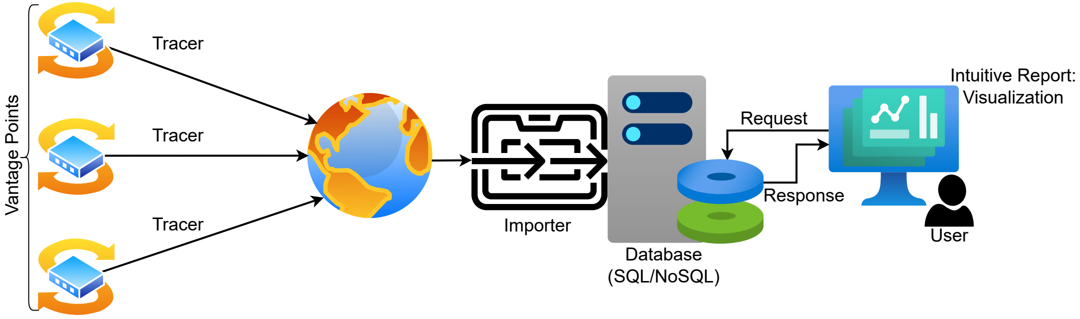


Figure 1: Overview of HiPerConTracer 3.0 architecture

It is worth noting that geo-location is part of the analysis, and not in the scope of the HiPerConTracer 3.0 measurements. For an analysis of own networks, e.g. the cloud/fog infrastructure and intranets of a service provider, precise geo-location data of own components would clearly be obtained from internal management systems rather than using approximations.

B. Route Tracing Analysis

Table II presents the results of an analysis of Traceroute data for 10 chosen servers (based on the number of countries observed). The mapping of observed links to countries and ASs is only possible and counted when both, the source and destination IP addresses of a link, are known (i.e., routers responded with ICMP `Time Exceeded`).

To make the data easier to read, the observed links, along with their AS mappings, have been visualized in Figure 2. Routers are represented by the country flag of their geo-location; end-systems are furthermore highlighted by a yellow rhombus as background. Different colors represent different ASs; solid lines indicate intra-AS links, while dashed lines show inter-AS links (the color is based on the *source* AS). This tartan-like pattern of the plots gave the HiPerConTracer development branch its code name “TARTAN”. Note again that links are only visible if source and destination routers are known. We did *not* attempt to *guess* the missing links to illustrate the information gaps. The line thickness corresponds to the percentage of the occurrences of a link in the Traceroute measurements between the source and destination. Due to the density of routers and links within the various small countries of Europe (dark green rectangle), the bottom part of the figure magnifies this region for better visibility.

An interesting result of this example is that 13 countries were found on the route to Athens, 🇬🇷 Greece (ftp.ntua.gr) with a geographically shortest distance of only approximately 2100 km. This list includes countries from the European Union (EU) and European Economic Area (EEA), and the United Kingdom (which is not a member of either EU or EEA). In terms of networks, 6 different ASs were involved. When looking at other EU destinations such as 🇧🇬 Bulgaria (ftp.uni-sofia.bg), 🇭🇷 Croatia (www.fesb.unist.hr) and 🇭🇺 Hungary (mirror.szerverem.hu), all the routes include the United Kingdom, i.e. a non-EU/non-EEA country. Another noteworthy result is the connectivity to Buenos Aires, 🇦🇷 Argentina (mirror.fcaglp.unlp.edu.ar), which involves 10 different ASs in 16 countries. The list of countries includes the United Kingdom, 🇧🇷 Brazil, 🇵🇦 Panama, and the 🇺🇸 United States. This means that, with regard to privacy, the observed routes of the measurement contain some unexpected elements: significantly more countries, regions, and network operators (i.e. ASs) may be involved than what a user may expect.

C. RTT Measurements Analysis

The HiPerConTracer 3.0 measurement data not only includes the routes, but also RTTs from the measurement system to each hop (i.e., from the vantage point in Oslo to the routers and the destination system itself). Figure 3 displays the average RTT in the form of a map plot, similar to the AS plot in Figure 2, to illustrate the latencies. More blue the link color, lower the latency; more red the link color, higher the latency. It is evident that links far away from Oslo, 🇳🇴 Norway, have a higher RTT. Long-term RTT measurements provide the opportunity to detect concealed detours, e.g. due to Layer-2 transport of packets via MPLS. Since the speed of light sets a strict lower limit for the signal propagation delay, any detour must have a higher minimum latency than a direct route.

To further visualise the changes over time, Figure 4 presents the cumulative distribution function (CDF) of the RTTs for the selected destinations, distinguished between IPv4 and IPv6: the y-axis shows the fraction of RTT recordings $\leq$ the corresponding RTT value on the x-axis. As expected, for each destination, there are a few small values (the achievable minimum in the ideal case), and a small spread for higher values (cases of high load). Of most interest are the observable steps, e.g. for the IPv4 relation to 🇮🇷 Iran (ctan.yazd.ac.ir) or the IPv6 relations to 🇬🇷 Greece (ftp.ntua.gr; small steps) and 🇮🇩 Indonesia (cran.usk.ac.id; larger steps). These denote changes of the routing, leading to changed latency of the underlying routes. That is, the data packets may take detours. Over time, the RTT changes due to different routes.

D. Hop Count Analysis

Latency changes can be caused by detours occurring due to rerouting, e.g. due to link failures or changes in network costs leading to new routes. This is usually visible on the IP layer by the minimum hop count (i.e. the required TTL/Hop Limit setting) needed to reach a destination. Table III presents

Table II: Countries and Autonomous Systems

Name	Location	IP	#C	Countries	#AS
cran.ncc.metu.edu.tr	Turkey	IPv4	12	Austria, Czechia, Denmark, France, Germany, Hungary, Netherlands, Norway, Slovakia, Switzerland, Turkey, United Kingdom	6
cran.usk.ac.id	Indonesia	IPv6	11	China, Denmark, Germany, Hong Kong, Indonesia, Netherlands, Norway, Singapore, Sweden, United Kingdom, United States	10
ctan.yazd.ac.ir	Iran	IPv4	13	Azerbaijan, Belgium, Denmark, Germany, Hungary, Iran, Italy, Netherlands, Norway, Russia, Sweden, Turkey, United Kingdom	11
ftp.ntua.gr	Greece	IPv6	13	Austria, Belgium, Czechia, Denmark, France, Germany, Greece, Italy, Netherlands, Norway, Sweden, Switzerland, United Kingdom	6
ftp.uni-sofia.bg	Bulgaria	IPv4	13	Austria, Bulgaria, Croatia, Czechia, Denmark, France, Germany, Hungary, Netherlands, Norway, Romania, Switzerland, United Kingdom	5
ftp.uni-sofia.bg	Bulgaria	IPv6	15	Austria, Belgium, Bulgaria, Croatia, Czechia, Denmark, France, Germany, Hungary, Netherlands, Norway, Romania, Sweden, Switzerland, United Kingdom	6
mirror.fcaglp.unlp.edu.ar	Argentina	IPv4	16	Argentina, Brazil, Chile, Czechia, Denmark, France, Germany, Netherlands, Norway, Panama, Portugal, Spain, Sweden, Switzerland, United Kingdom, United States	10
mirror.szerverem.hu	Hungary	IPv4	13	Austria, Croatia, Czechia, Denmark, France, Germany, Hungary, Netherlands, Norway, Romania, Slovakia, Switzerland, United Kingdom	4
mirrors.cqu.edu.cn	China	IPv4	11	China, Denmark, France, Germany, Hong Kong, Netherlands, Norway, Spain, Sweden, United Kingdom, United States	5
www.fesb.unist.hr	Croatia	IPv4	12	Austria, Croatia, Czechia, Denmark, France, Germany, Hungary, Netherlands, Norway, Slovakia, Switzerland, United Kingdom	4

Table III: Hop count statistics for 10 destinations over three months.

Name	Location	IP	Hops.Min	Hops.$Q_{05\%}$	Hops.Mean	Hops.Median	Hops.$Q_{95\%}$
cran.ncc.metu.edu.tr	Turkey	IPv4	15	24.0	24.0	24.0	24.0
cran.usk.ac.id	Indonesia	IPv6	18	18.0	20.6	21.0	21.0
ctan.yazd.ac.ir	Iran	IPv4	20	21.0	21.3	21.0	23.0
ftp.ntua.gr	Greece	IPv6	21	21.0	23.0	23.0	23.0
ftp.uni-sofia.bg	Bulgaria	IPv4	23	23.0	24.7	25.0	25.0
ftp.uni-sofia.bg	Bulgaria	IPv6	24	24.0	25.7	26.0	26.0
mirror.fcaglp.unlp.edu.ar	Argentina	IPv4	26	27.0	27.2	27.0	28.0
mirror.szerverem.hu	Hungary	IPv4	21	22.0	22.0	22.0	22.0
mirrors.cqu.edu.cn	China	IPv4	24	24.0	24.6	25.0	26.0
www.fesb.unist.hr	Croatia	IPv4	21	22.0	22.7	23.0	24.0

the hop count statistics for the 10 destinations over the three months of measurement, including the minimum, mean, and 95% quantile hop count. The mean hop count is usually slightly higher than the minimum, indicating that there is sometimes a slightly longer route. However, the difference for the destination in Turkey (cran.ncc.metu.edu.tr) is much larger: 24 vs. 15. This means that the length of a route can change drastically, which may come as a surprise to users and application developers. Applications must be prepared to handle this behavior to adhere to their service-level requirements.

E. Applicability

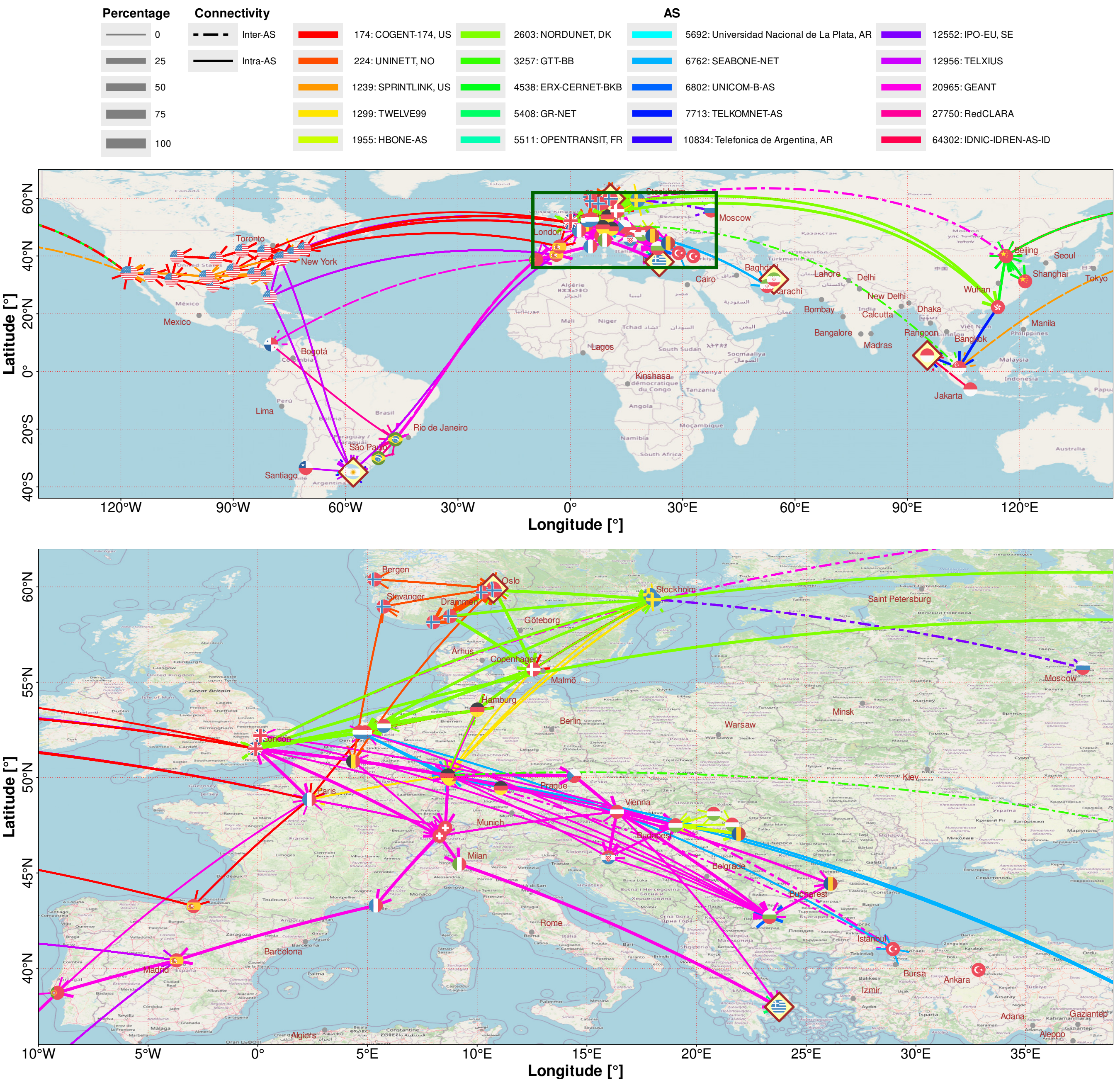
To improve congestion, adaptive transmission rate, latency, and jitter, both short- and long-term latency variations must be considered. HiPerConTracer can help to obtain accurate information about the underlying networks and customize the application parameters. This is particularly also of interest for 5G core/edge infrastructures, high-quality video streaming, and virtual reality.

V. Conclusion and Future Work

As a proof of concept, we demonstrate long-term observation of Internet server connectivity. The results demonstrate that the route length can change drastically without a clear explanation. During our experiments, we observed that routes can contain unexpected countries, regions, and network operators. Such packet routing analysis can be used to monitor data packet flows and assess the status of the infrastructure (e.g. frequent detours or packet loss). Ultimately, it can help to identify performance problems in network locations.

Our future work plan is to extend HiPerConTracer 3.0 by adding new protocol modules, measurements, new data management features, and more feature-rich analyses.

Figure 2: Visualization of AS mapping for observed links.



Source Code and Data Availability

The source code of HiPerConTracer 3.0 is available from https://github.com/dreibh/hipercontracer/tree/tartan under GNU General Public License version 3.0. The data set of the presented results is also available via IEEE DataPort at https://ieee-dataport.org/documents/tartan-traceroute-dataset (DOI 10.21227/a241-gm35).

References

[1] R. R. Stewart, "Stream Control Transmission Protocol," IETF, RFC 4960, Sep. 2007, doi:10.17487/RFC4960.

[2] J. B. Postel, "Internet Control Message Protocol," IETF, RFC 792, Sep. 1981, doi:10.17487/RFC0792.

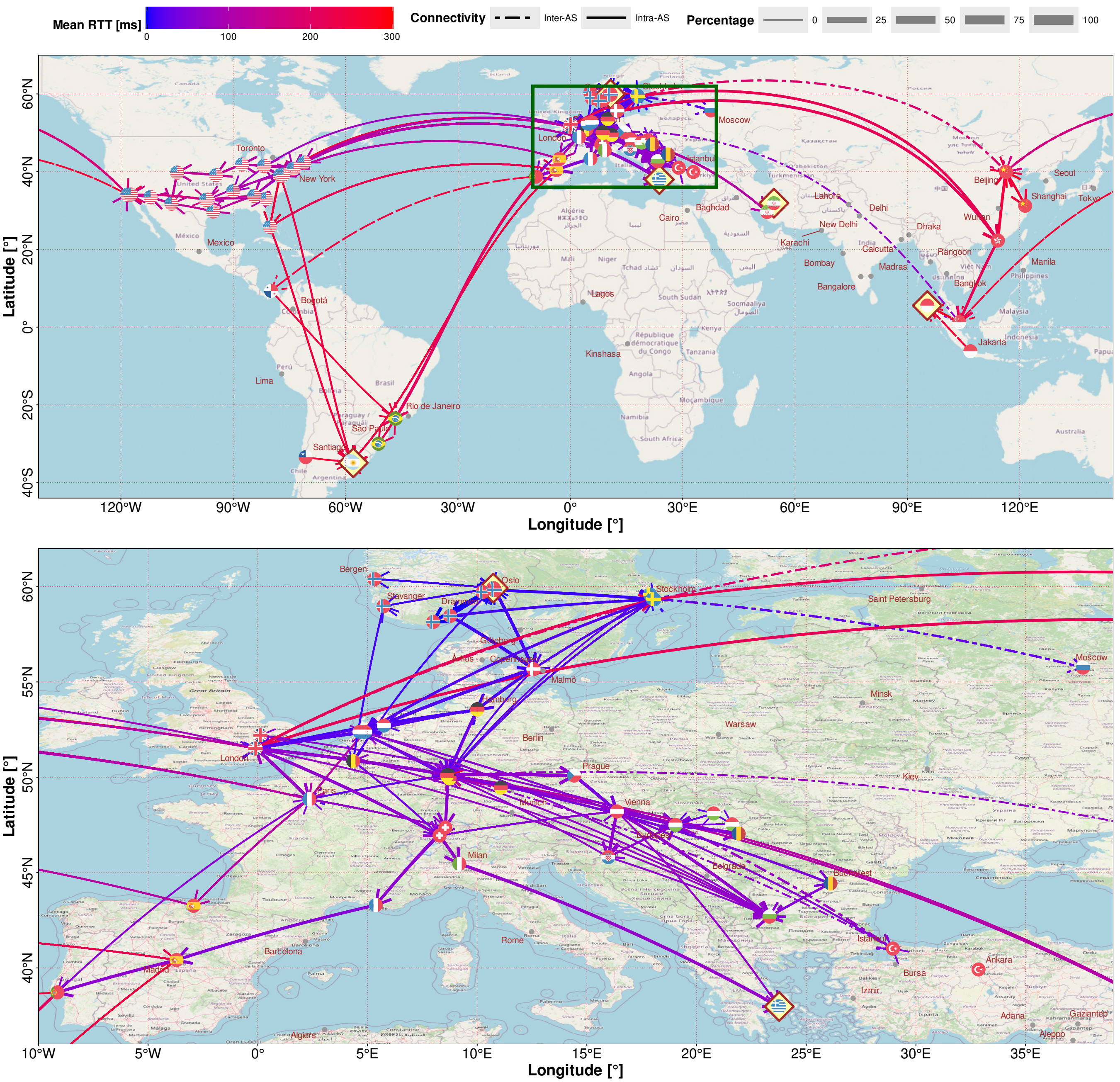
[3] A. Conta, S. E. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," IETF, Standards Track RFC 4443, Mar. 2006, doi:10.17487/RFC4443.

[4] J. B. Postel, "Internet Protocol," IETF, RFC 791, Sep. 1981, doi:10.17487/RFC0791.

[5] S. E. Deering and R. M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," IETF, Standards Track RFC 2460, Dec. 1998, doi:10.17487/RFC2460.

[6] M. Maier and J. Ullrich, "In the Loop: A Measurement Study of Persistent Routing Loops on the IPv4/IPv6 Internet," *Computer Networks*, vol. 221, p. 109500, 2023, doi:10.1016/j.comnet.2022.109500.

Figure 3: Observed latencies: Average RTT mapping for observed links



[7] J.-R. Luttringer, Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet, “Let There be Light: Revealing Hidden MPLS Tunnels with TNT,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1239–1253, 2019, doi:10.1145/2185376.2185388.

[8] P. McCherry, V. Giotsas, and D. Hutchison, “On Improving the Accuracy of Internet Infrastructure Mapping,” *IEEE Access*, vol. 11, pp. 59 935–59 953, 2023, doi:10.1109/ACCESS.2023.3281333.

[9] Y. Huang, M. Rabinovich, and R. Al-Dalky, “FlashRoute: Efficient Traceroute on a Massive Scale,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20. Virtual Event: Association for Computing Machinery, 2020, pp. 443–455, doi:10.1145/3419394.3423619.

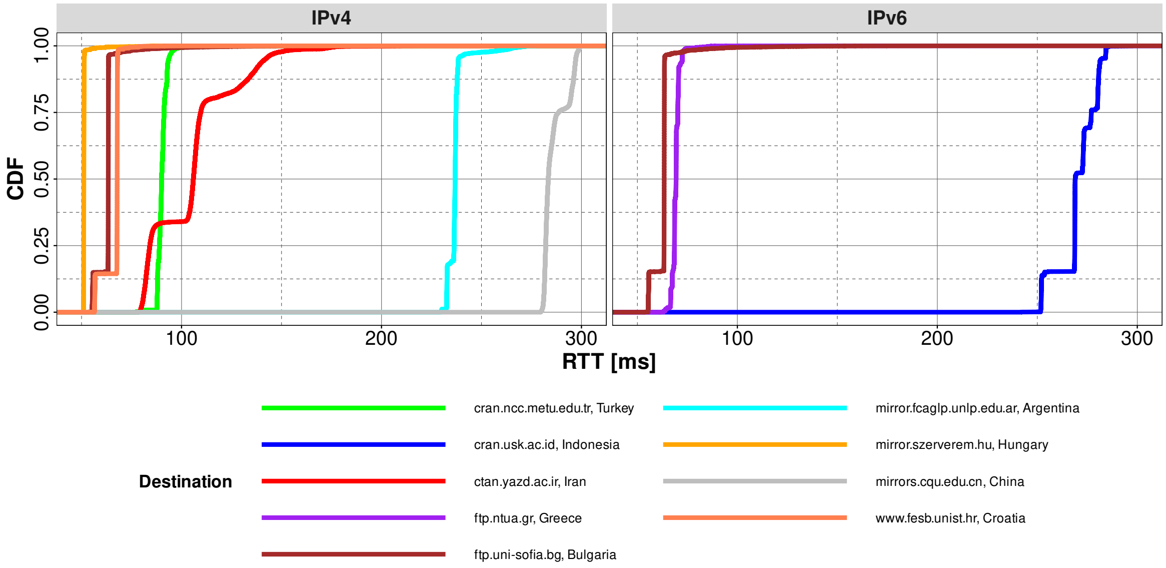
[10] T. Dreibholz, “HiPerConTracer - A Versatile Tool for IP Connectivity Tracing in Multi-Path Setups,” in *Proceedings of the 28th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar, Dalmacija/Croatia, Sep. 2020, pp. 1–6, doi:10.23919/SoftCOM50211.2020.9238278.

[11] ——, “High-Precision Round-Trip Time Measurements in the Internet with HiPerConTracer,” in *Proceedings of the 31st International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Dalmacija/Croatia, Sep. 2023, doi:10.23919/SoftCOM58365.2023.10271612.

[12] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, “10 Lessons from 10 Years of Measuring and Modeling the Internet’s Autonomous Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1810–1821, 2011, doi:10.1109/JSAC.2011.111006.

[13] R. Singh, A. Dunna, and P. Gill, “Characterizing the Deployment and Performance of Multi-CDNS,” in *Proceedings of*

Figure 4: CDF of RTTs for selected destinations, covering IPv4 and IPv6.



the Internet Measurement Conference (IMC), 2018, pp. 168–174, doi:10.1145/3278532.3278548.

[14] J. B. Postel, “Echo Protocol,” IETF, RFC 862, May 1983, doi:10.17487/RFC0862.

[15] J. M. Evang and T. Dreibholz, “Optimizing Network Latency: Unveiling the Impact of Reflection Server Tuning,” in *Proceedings of the 6th International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC)*, Apr. 2024, doi:10.1007/978-3-031-57942-4_36.

[16] J. B. Postel, “Transmission Control Protocol,” IETF, RFC 793, Sep. 1981, doi:10.17487/RFC0793.

[17] W. M. Eddy, “TCP SYN Flooding Attacks and Common Mitigations,” IETF, Informational RFC 4987, Aug. 2007, doi:10.17487/RFC4987.

[18] M. Allman, V. Paxson, and E. Blanton, “TCP Congestion Control,” IETF, Standards Track RFC 5681, Sep. 2009, doi:10.17487/RFC5681.

[19] J. B. Postel, “User Datagram Protocol,” IETF, RFC 768, Aug. 1980, doi:10.17487/RFC0768.

[20] Q. Scheitle, O. Gasser, P. Sattler, and G. Carle, “HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks,” in *Proceedings the of Network Traffic Measurement and Analysis Conference (TMA)*, Dublin/Ireland, Jun. 2017, pp. 1–9, doi:10.23919/TMA.2017.8002903.