SC23
Denver, CO | i am hpc.

Evaluation of SCION for User-driven Path Control: a Usability Study

Antonio Battipaglia, Leonardo Boldrini, Ralph Koning, Paola Grosso

UPIN

Politecnico di Torino

University of Amsterdam

Responsible Internet

The **Responsible Internet** [1] is a novel security-by-design concept and extension to the internet that enables higher levels of trust and data autonomy.

It turns the Internet infrastructure from a black box to a ‘glass box’

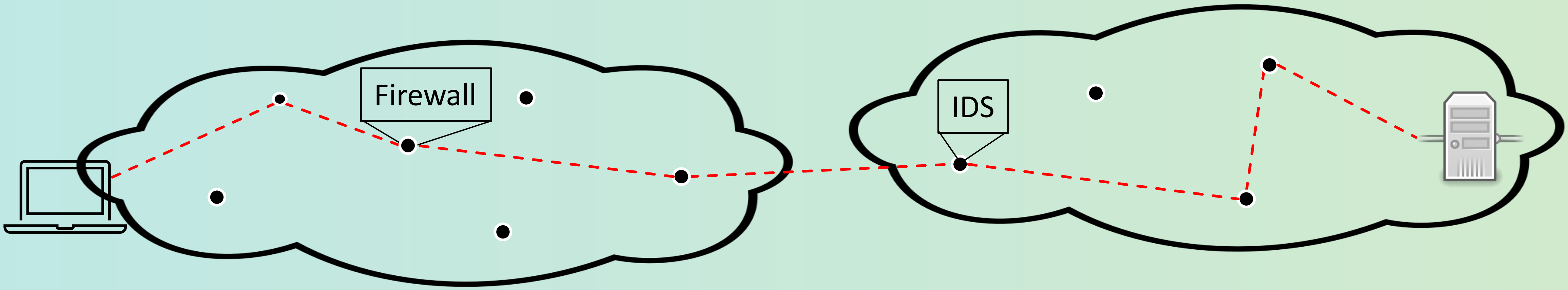
It brings Transparency, Controllability and Accountability to the Internet

[1] Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane Moura, et al. **A responsible internet to increase trust in the digital world**. 2020.

The UPIN project

UPIN: User-driven Path verification and control for Inter-domain Networks enables users to control and verify paths that their data travels through [2]

Path characteristics include locations, manufacturers, jurisdictions applied, VNFs, …



[2] Rodrigo Bazo, Leonardo Boldrini, Cristian Hesselman, and Paola Grosso. **Increasing the Transparency, Accountability and Controllability of multi-domain networks with the UPIN framework**. 2021

SCION

SCION [3] is an Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications to end users.

SCION

SCION [3] is an Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications to end users.

SCION organizes existing ASes into groups of independent routing planes, called isolation domains (**ISD**s), which interconnect to provide global connectivity. Isolation domains provide natural isolation of routing failures and misconfigurations, **give endpoints strong control** for both inbound and outbound traffic.

SCION

SCION [3] is an Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications to end users.

SCION organizes existing ASes into groups of independent routing planes, called isolation domains (**ISD**s), which interconnect to provide global connectivity. Isolation domains provide natural isolation of routing failures and misconfigurations, **give endpoints strong control** for both inbound and outbound traffic.

As a path-based architecture, SCION end hosts learn about available network path segments, and combine them into end-to-end paths that are carried in packet headers. This approach enables **path-aware communication**, an emerging trend in networking.

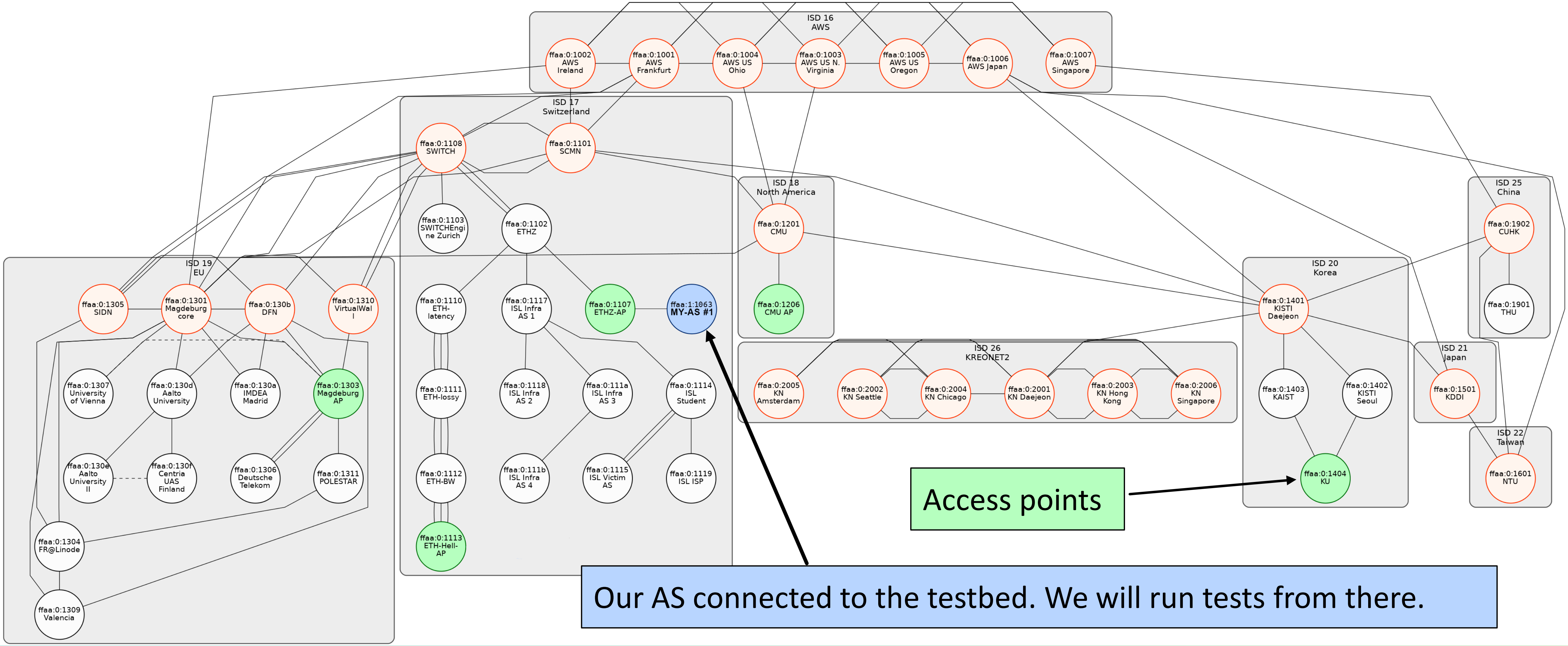
[3] Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig, and David G. Andersen. **SCION: Scalability, Control, and Isolation on Next-Generation Networks**. 2011

Our testbed: SCIONLab

SCIONLab provides a fully distributed SCION network infrastructure, made up by different ASes organized in ISDs.

Researchers can define their own ASes and connect them to the SCION network, for running experiments.

Our testbed: SCIONLab



Goals and experiments

- We want to provide UPIN users with the ability to select which paths their traffic takes

Goals and experiments

- We want to provide UPIN users with the ability to select which paths their traffic takes
- We analyze SCION features of path awareness and path selection

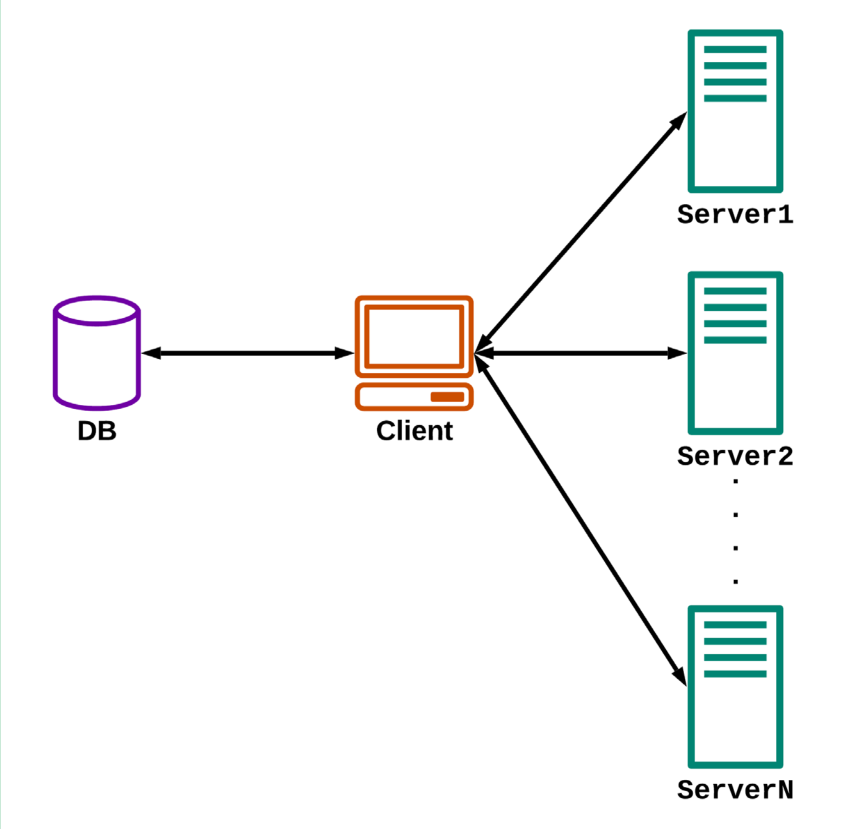
Goals and experiments

- We want to provide UPIN users with the ability to select which paths their traffic takes
- We analyze SCION features of path awareness and path selection
- We test network performance of **latency, bandwidth and packet loss** in order to provide UPIN users with paths that fulfill requirements on these properties

Goals and experiments

- We want to provide UPIN users with the ability to select which paths their traffic takes
- We analyze SCION features of path awareness and path selection
- We test network performance of **latency, bandwidth and packet loss** in order to provide UPIN users with paths that fulfill requirements on these properties

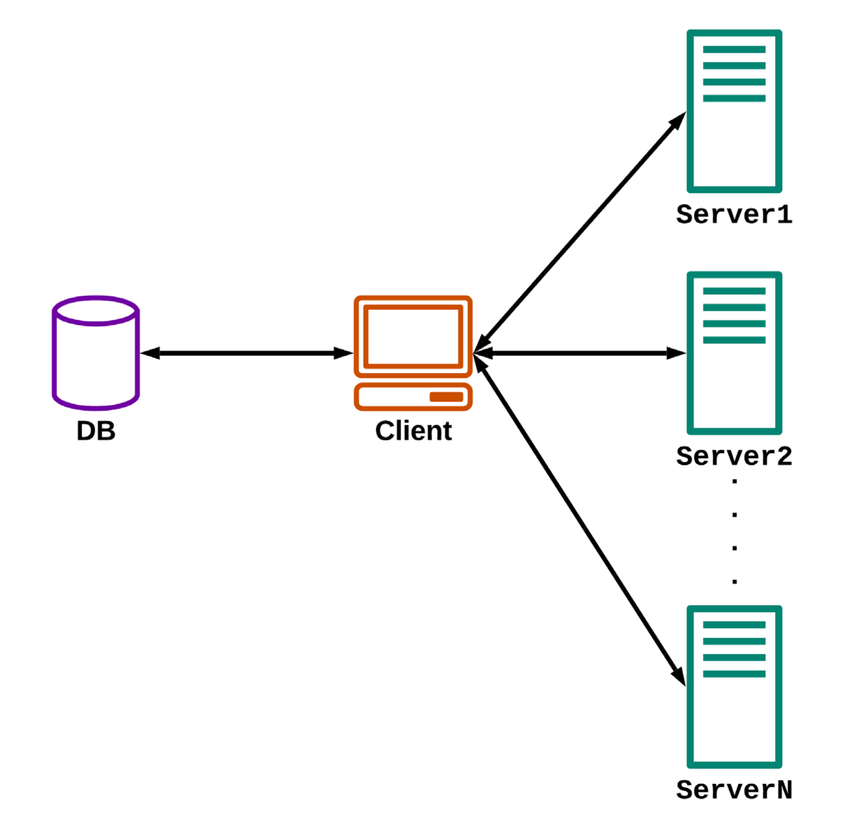
- This is an overview of our 3-tier software architecture
  - Paths collection: we gather what paths are available from our client to each destination or server
  - Path test execution: we test each path performance
  - Stats storage: we store data in a database



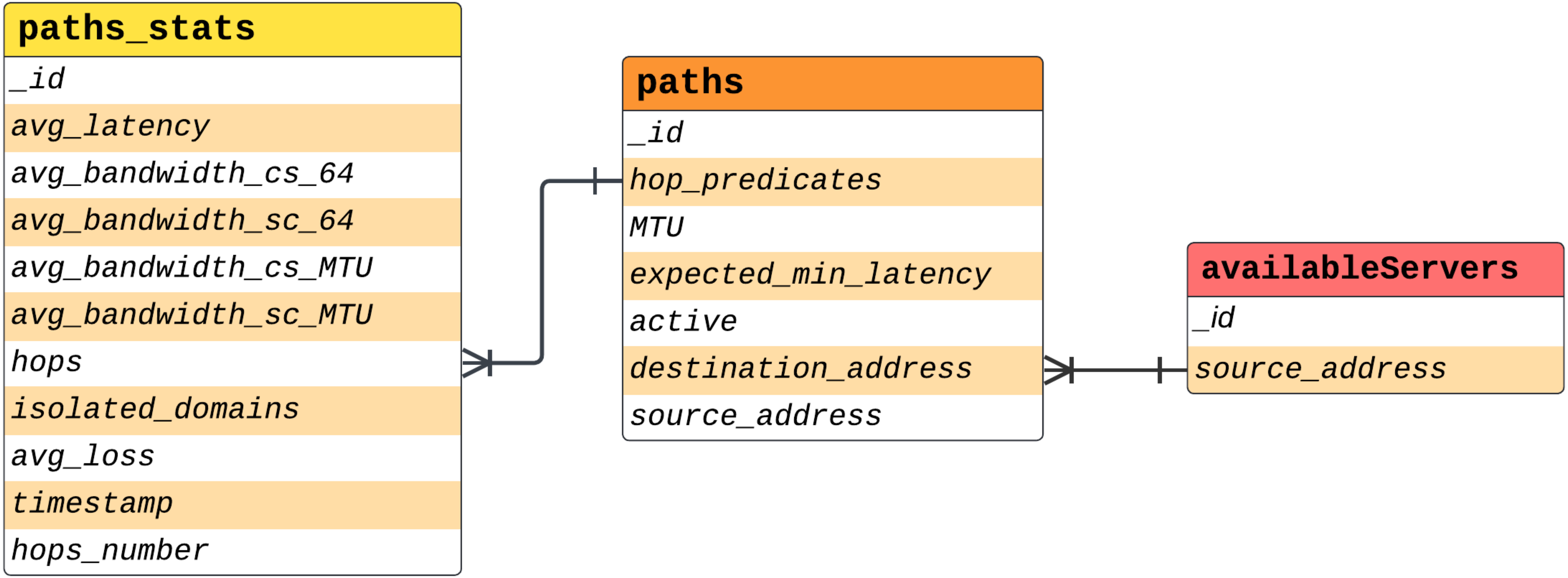
Goals and experiments

- We want to provide UPIN users with the ability to select which paths their traffic takes
- We analyze SCION features of path awareness and path selection
- We test network performance of **latency, bandwidth and packet loss** in order to provide UPIN users with paths that fulfill requirements on these properties

- This is an overview of our 3-tier software architecture
  - Paths collection: we gather what paths are available from our client to each destination or server
  - Path test execution: we test each path performance
  - Stats storage: we store data in a database
- Then we can query our DB to provide users with the path they need



Database structure



Implementation

SCIONLab offer a set of commands useful for us to run our experiments:

Implementation

SCIONLab offer a set of commands useful for us to run our experiments:

- scion showpaths −−extended −m 40

  This command provides a maximum of 40 paths for each destination, ranked by hop count, along with hops predicates (hops traversed in the path), MTU, path status and minimum latency expected over the path. We use this command for paths collection.

Implementation

SCIONLab offer a set of commands useful for us to run our experiments:

- scion showpaths −−extended −m 40

  This command provides a maximum of 40 paths for each destination, ranked by hop count, along with hops predicates (hops traversed in the path), MTU, path status and minimum latency expected over the path. We use this command for paths collection.

- scion ping {server_address} −c 30 −−sequence ‘{hop_predicates}' −−interval 0.1s

  From the output of this command we calculate the average latency measured by the 30 packets sent in milliseconds and the packet loss percentage. We run this command in 3 nested loops: per each path, per each destination, ‘iteration’ number of times.

Implementation

SCIONLab offer a set of commands useful for us to run our experiments:

- scion showpaths −−extended −m 40

  This command provides a maximum of 40 paths for each destination, ranked by hop count, along with hops predicates (hops traversed in the path), MTU, path status and minimum latency expected over the path. We use this command for paths collection.

- scion ping {server_address} −c 30 −−sequence ‘{hop_predicates}' −−interval 0.1s

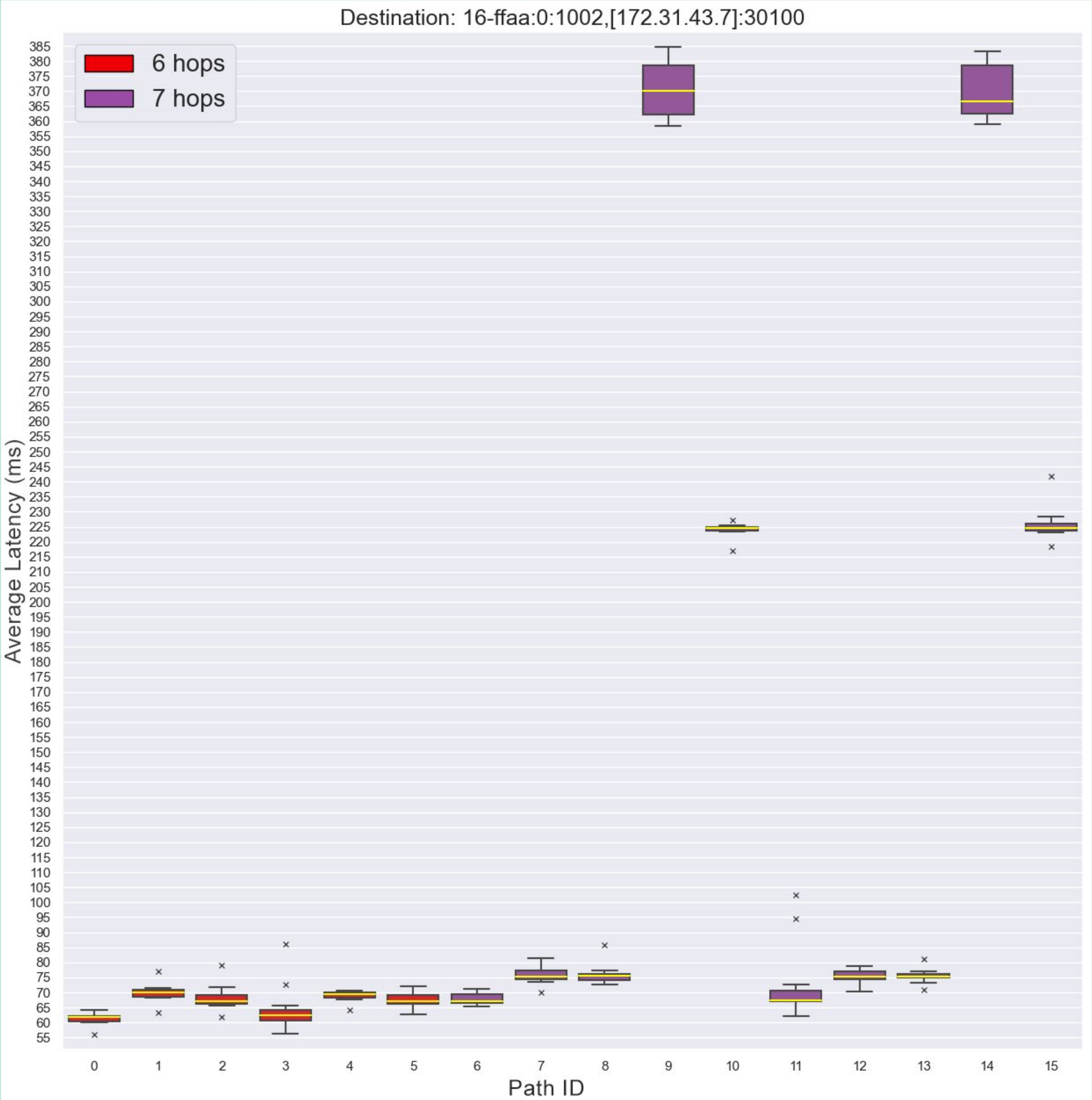
  From the output of this command we calculate the average latency measured by the 30 packets sent in milliseconds and the packet loss percentage. We run this command in 3 nested loops: per each path, per each destination, ‘iteration’ number of times.

- scion−bwtestclient −s {server_address} −cs 3,64,?,12Mbps −sequence '{hop_predicates}’

  For the bandwidth we add the test duration (3s), the packet size (64 bytes), a wildcard for the number of packets automatically computed by the application, and the desired bandwidth to achieve (in this case 12Mbps). Downlink and uplink are saved as different values.

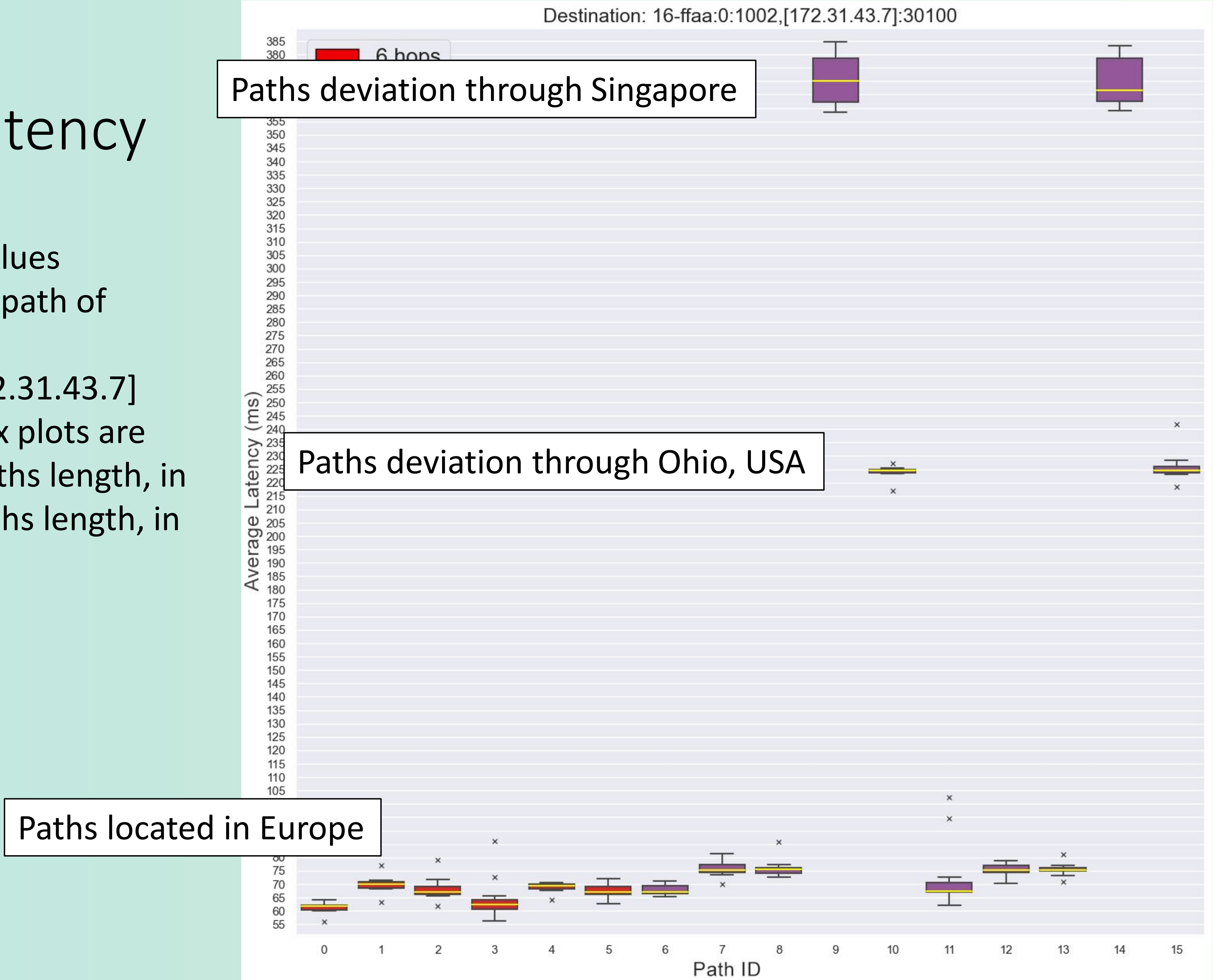
Results: Latency

Average Latency Values measured for each path of destination 16-ffaa:0:1002,[172.31.43.7] (AWS - Ireland). Box plots are split into 6 hops paths length, in red, and 7 hops paths length, in purple.



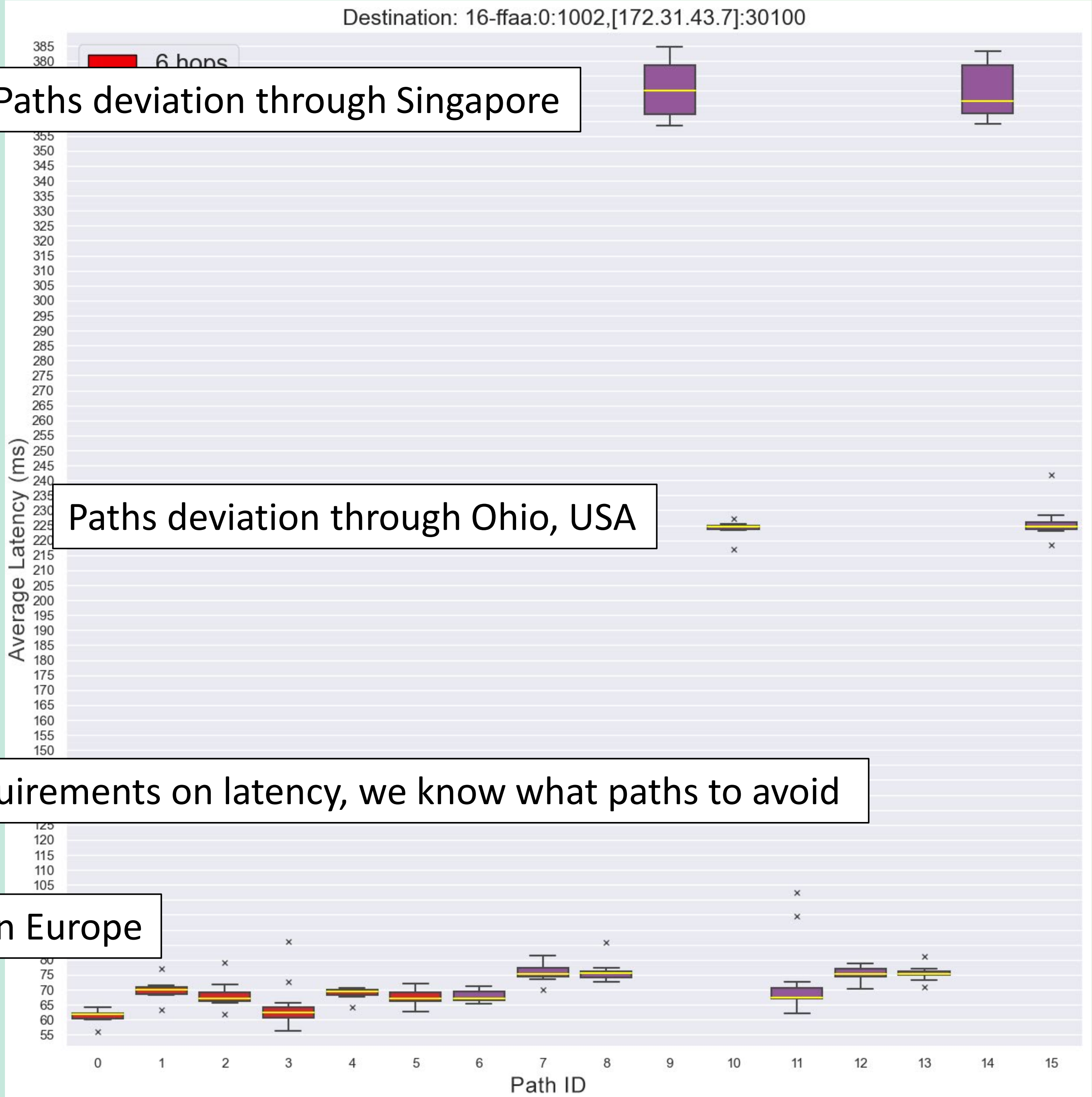
Results: Latency

Average Latency Values measured for each path of destination 16-ffaa:0:1002,[172.31.43.7] (AWS - Ireland). Box plots are split into 6 hops paths length, in red, and 7 hops paths length, in purple.



Results: Latency

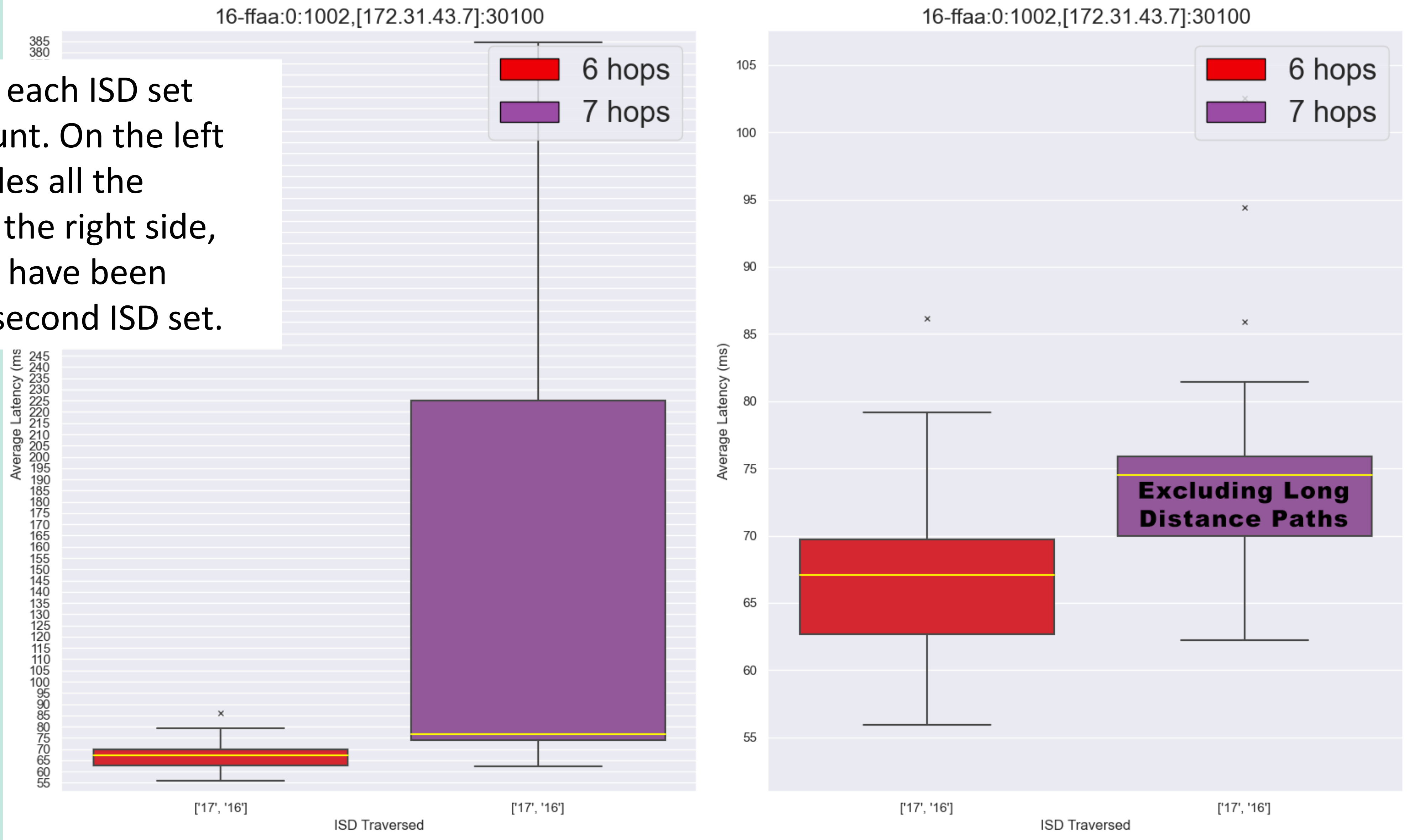
Average Latency Values measured for each path of destination 16-ffaa:0:1002,[172.31.43.7] (AWS - Ireland). Box plots are split into 6 hops paths length, in red, and 7 hops paths length, in purple.



If we need to satisfy strong requirements on latency, we know what paths to avoid

Results: Latency

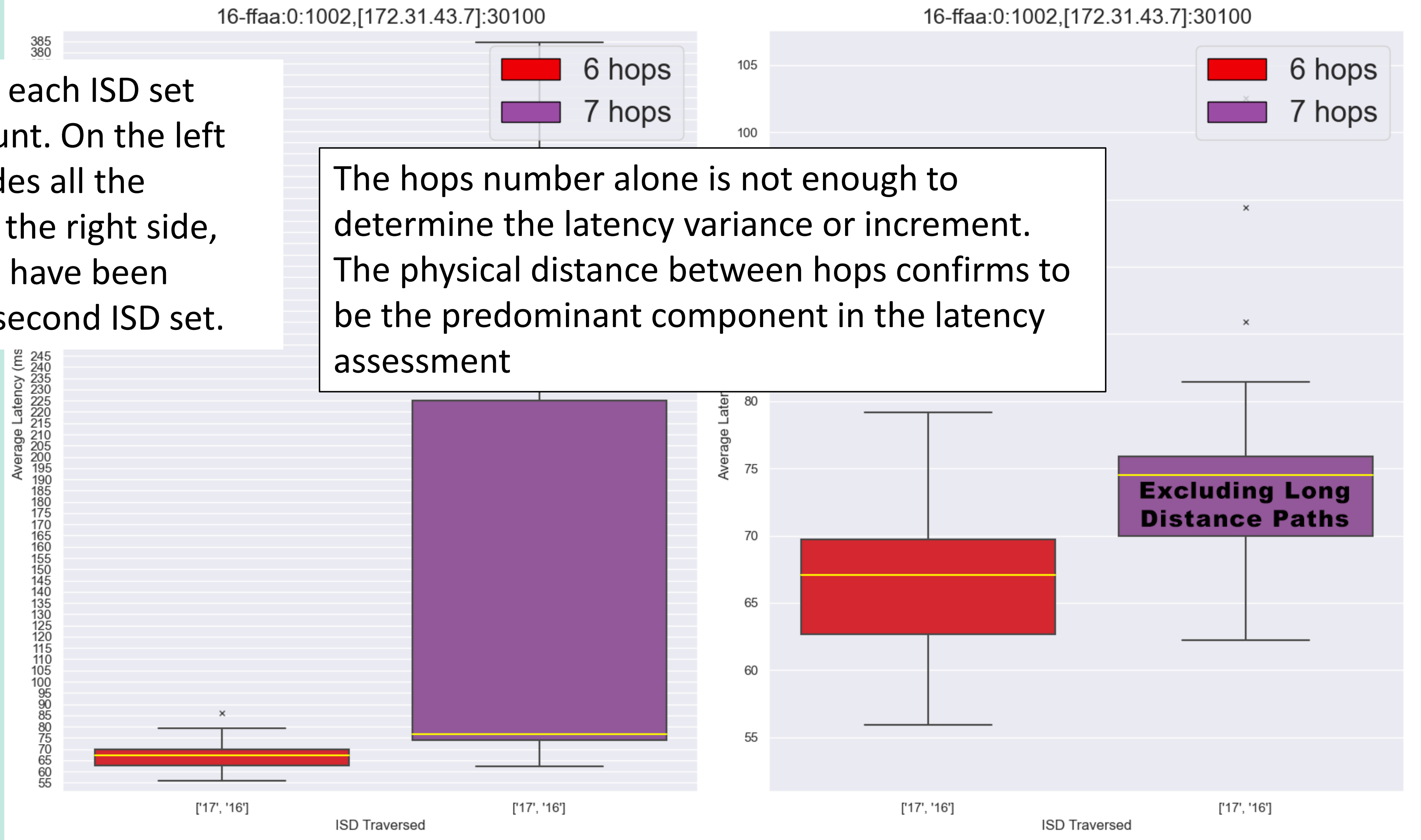
Average latency for each ISD set grouped by hop count. On the left side, the plot includes all the measurements. On the right side, long distance paths have been excluded from the second ISD set.



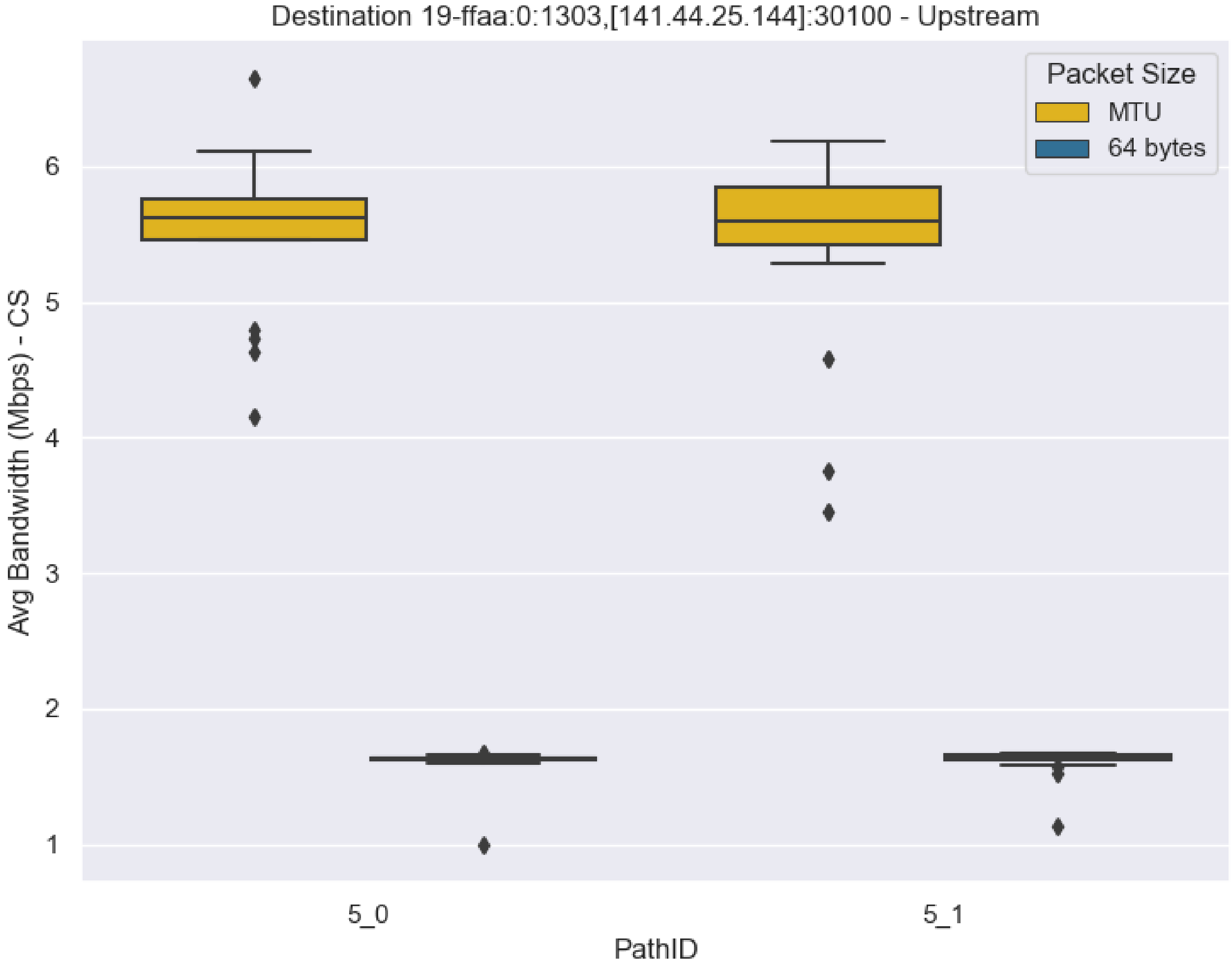
Results: Latency

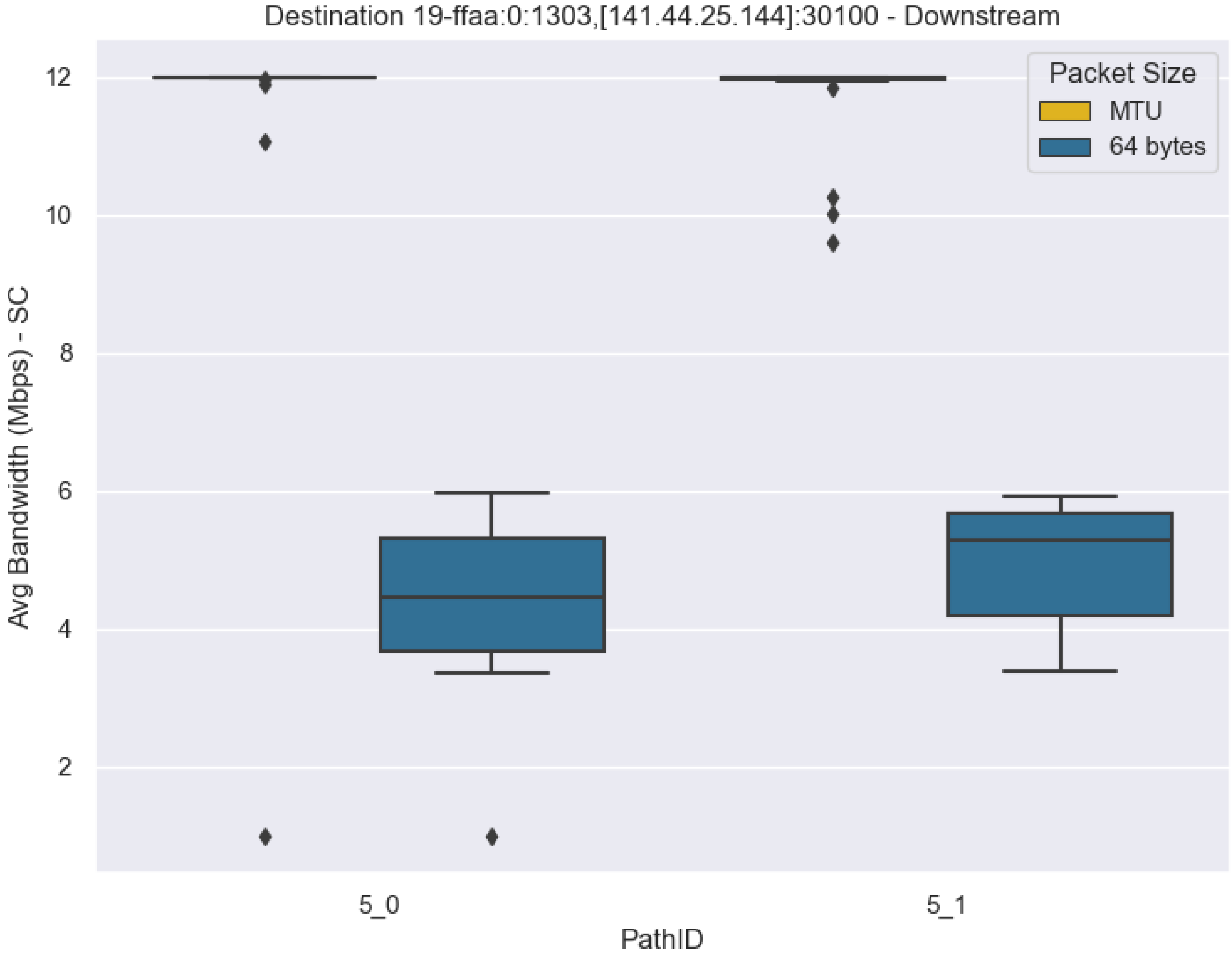
Average latency for each ISD set grouped by hop count. On the left side, the plot includes all the measurements. On the right side, long distance paths have been excluded from the second ISD set.

The hops number alone is not enough to determine the latency variance or increment. The physical distance between hops confirms to be the predominant component in the latency assessment



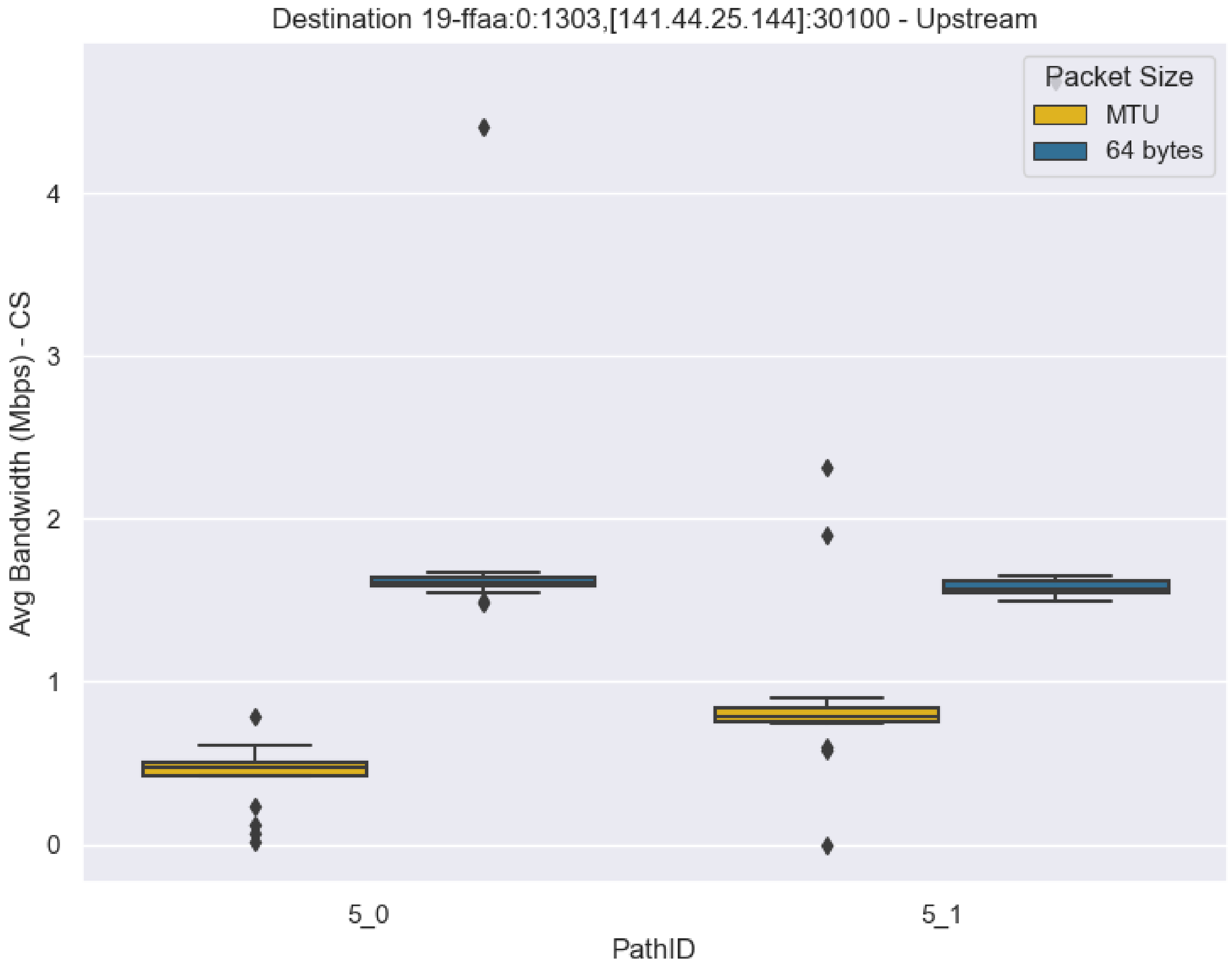
Results: Bandwidth

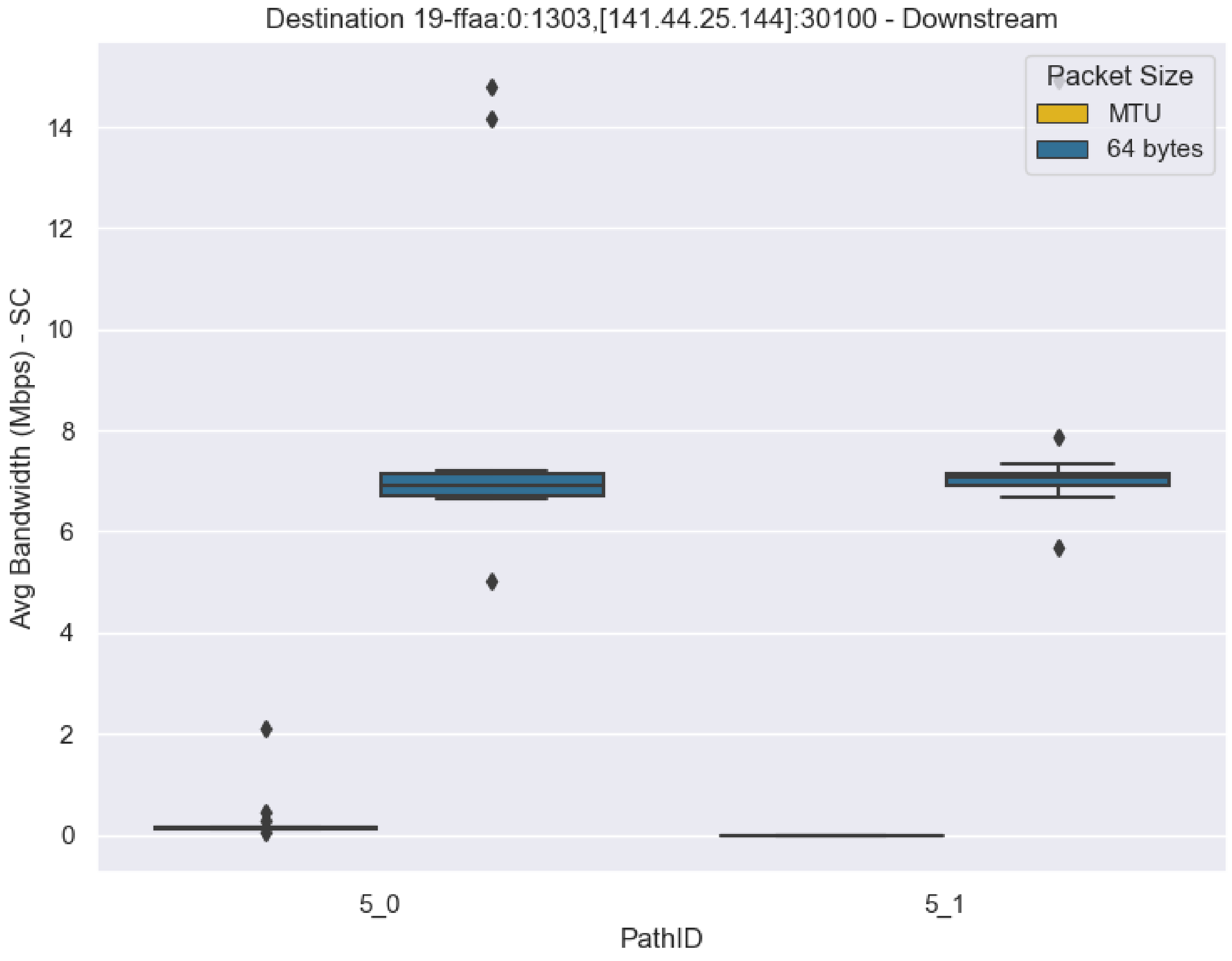




- Average bandwidth values for each path, requiring a bandwidth of 12Mbps from and to a Server in Germany (address on the top). On the left side there are the upstream measurements, while on the right side the downstream ones.

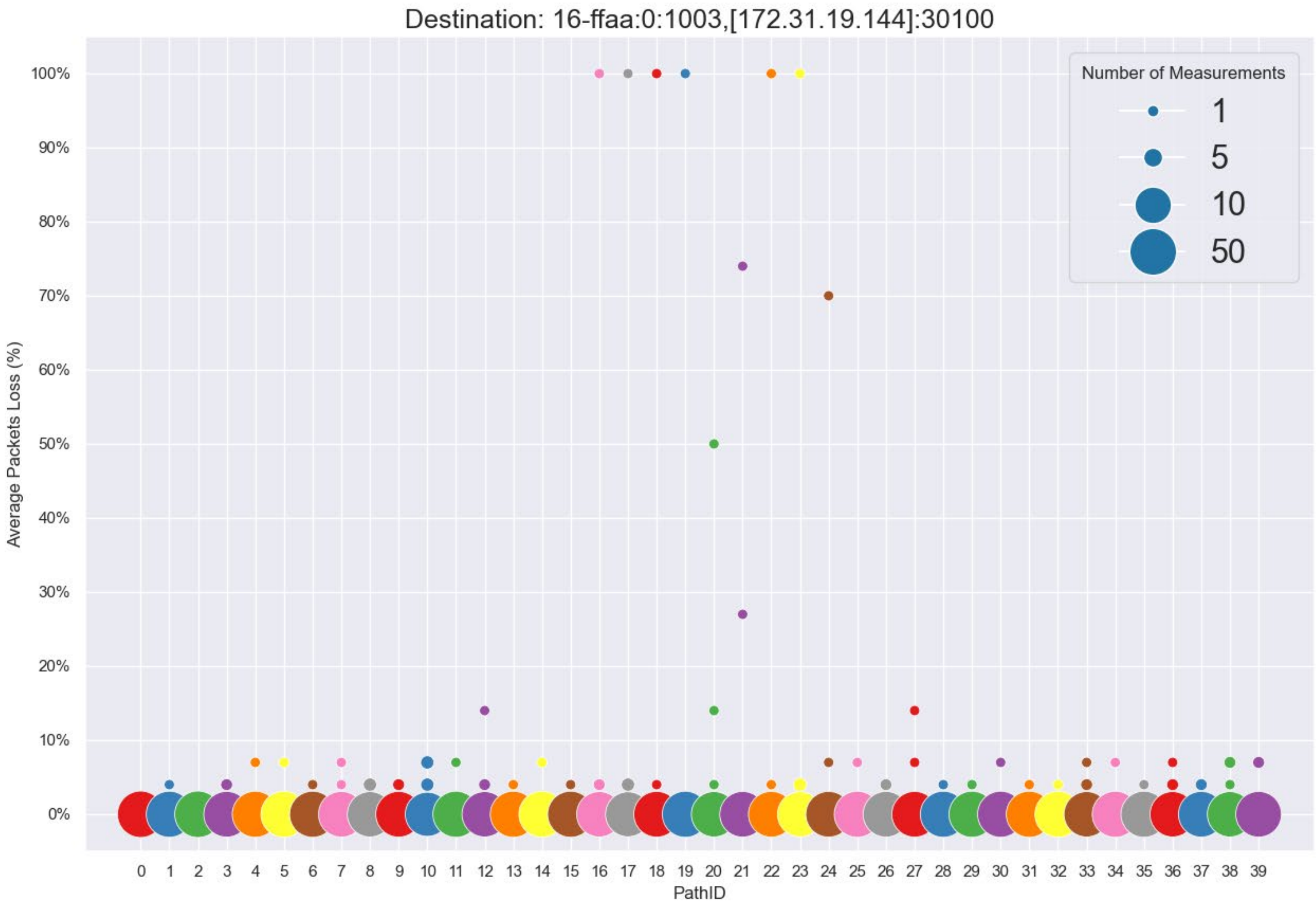
Results: Bandwidth





- Average bandwidth values for each path, requiring a bandwidth of 150 Mbps from and to a Server in Germany (address on the top). On the left side there are the upstream measurements, while on the right side the downstream ones.

Results: Packet Loss



- Average packet loss percentage for each path of AWS US N. Virginia AS. Each dot color represents a path and its size the number of measurements having the same loss ratio. Dots legend is on the upper right corner.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.
- We investigated the possibilities and limitations of SCIONLab for user-driven path control. We explored several aspects of its **performance**, specifically focusing on latency, bandwidth and data loss.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.
- We investigated the possibilities and limitations of SCIONLab for user-driven path control. We explored several aspects of its **performance**, specifically focusing on latency, bandwidth and data loss.
- The software design is based on a **database** where we store data gathered on many paths, that we then query to select the best one to give to a user.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.
- We investigated the possibilities and limitations of SCIONLab for user-driven path control. We explored several aspects of its **performance**, specifically focusing on latency, bandwidth and data loss.
- The software design is based on a **database** where we store data gathered on many paths, that we then query to select the best one to give to a user.
- We confirmed that **latency** in SCIONLab is affected mostly by the physical distance among the nodes building the path, rather than and that the number of hops or the ISDs traversed.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.
- We investigated the possibilities and limitations of SCIONLab for user-driven path control. We explored several aspects of its **performance**, specifically focusing on latency, bandwidth and data loss.
- The software design is based on a **database** where we store data gathered on many paths, that we then query to select the best one to give to a user.
- We confirmed that **latency** in SCIONLab is affected mostly by the physical distance among the nodes building the path, rather than and that the number of hops or the ISDs traversed.
- We also encountered and documented limitations of the SCIONLab network, where the capacity decreases when trying to target a higher bandwidth from a path.

Conclusion

- In order to achieve **user-driver path control**, we need to know several properties of the underlying network and what paths we can build in it.
- We investigated the possibilities and limitations of SCIONLab for user-driven path control. We explored several aspects of its **performance**, specifically focusing on latency, bandwidth and data loss.
- The software design is based on a **database** where we store data gathered on many paths, that we then query to select the best one to give to a user.
- We confirmed that **latency** in SCIONLab is affected mostly by the physical distance among the nodes building the path, rather than and that the number of hops or the ISDs traversed.
- We also encountered and documented limitations of the SCIONLab network, where the capacity decreases when trying to target a higher bandwidth from a path.
- The path selection feature of SCION, when coupled with a robust test-suite and data analysis techniques, blends into a powerful tool that helps to fulfill the controllability requirement of a UPIN user

Thank you