Hole punching in the wild

Learnings from running libp2p hole punching in production, measured from vantage points across the globe.

FOSDEM 2023

Dennis Trautwein and Max Inden

TALK

About us

Dennis Trautwein

Research Engineer at Protocol Labs

Github: @dennis-tra

Twitter: @dtrautwein_eu

Web: https://dtrautwein.eu

Email: dennis@protocol.ai

Max Inden

Software Engineer at Protocol Labs

GitHub/Twitter/...: @mxinden

Web: https://max-inden.de

Email: mail@max-inden.de

Agenda

- Quick intro to libp2p
- The Problem – Firewalls and NATs
- The Solution – Hole Punching
- Measurement Campaign
- Next Steps

libp2p

- Peer-to-peer networking library
- One specification, many implementations (Go, JS, Rust, Nim, C++, Java, …)
- Low level features like encryption, authentication and hole punching
- High level features like DHT or Gossiping
- All you need to build peer-to-peer applications

LIBP2P

Motivation

Full connectivity among all nodes of a libp2p network despite NATs and Firewalls

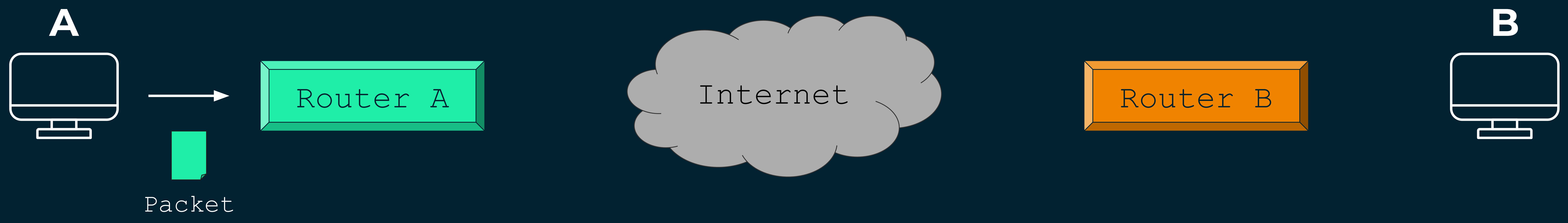
NATs and Firewalls

- **NAT**
  - Local to public IP address mapping
- **Firewall**
  - Control incoming/outgoing network traffic based on security rules

Source IP	Source Port	Dest. IP	Dest. Port	Transport
192.168.0.2	12345	198.51.100.0	54321	TCP
192.168.0.2	12345	198.51.100.1	54322	UDP
...	...	...	...	...

Hole Punching

Problem

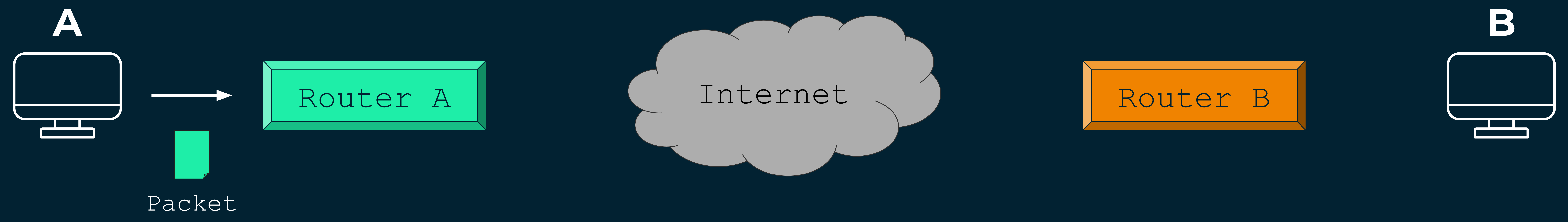


Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Hole Punching

Problem

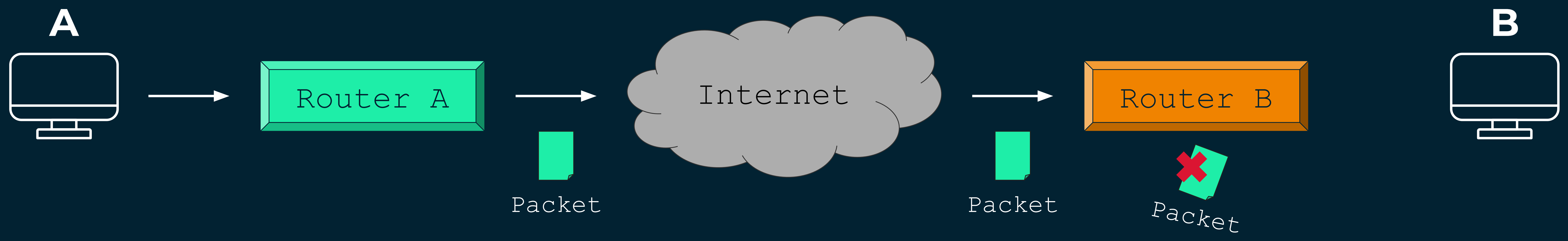


Src IP	Src Port	Dst IP	Dst Port	Trpt
Internal IP	Internal Port	Public IP Router B	Public Port Router B	TCP

Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Hole Punching

Problem



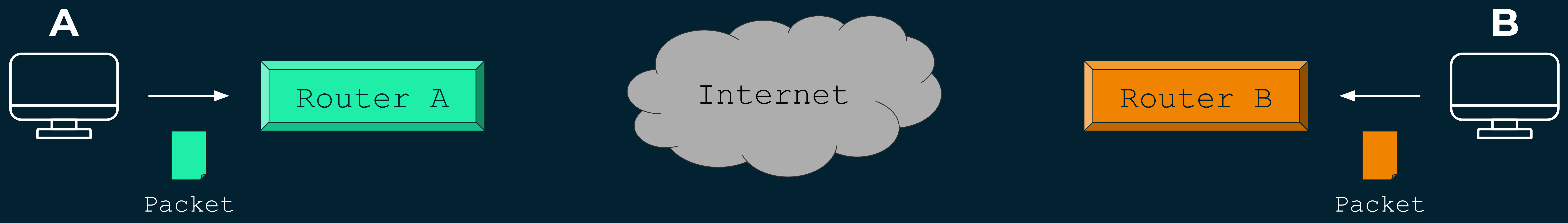
Src IP	Src Port	Dst IP	Dst Port	Trpt
Internal IP	Internal Port	Public IP Router B	Public Port Router B	TCP

Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Hole Punching

Hole Punching

Process

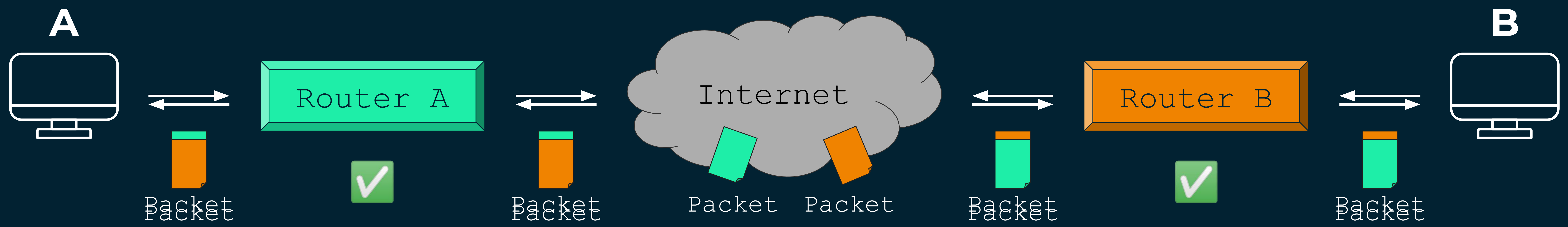


Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Src IP	Src Port	Dst IP	Dst Port	Trpt
–	–	–	–	–

Hole Punching

Process

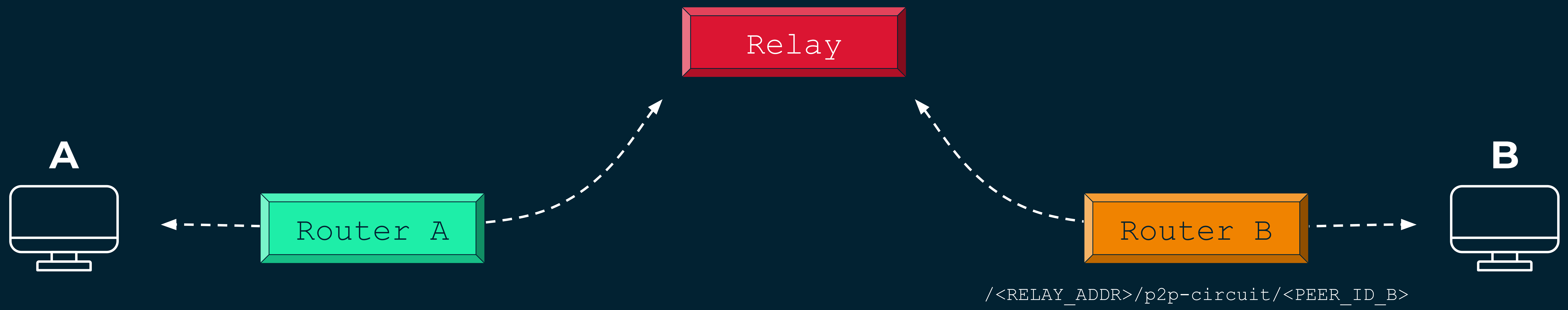


Src IP	Src Port	Dst IP	Dst Port	Trpt
Internal IP	Internal Port	Public IP Router B	Public Port Router B	TCP

Src IP	Src Port	Dst IP	Dst Port	Trpt
Internal IP	Internal Port	Public IP Router A	Public Port Router A	TCP

Hole Punching

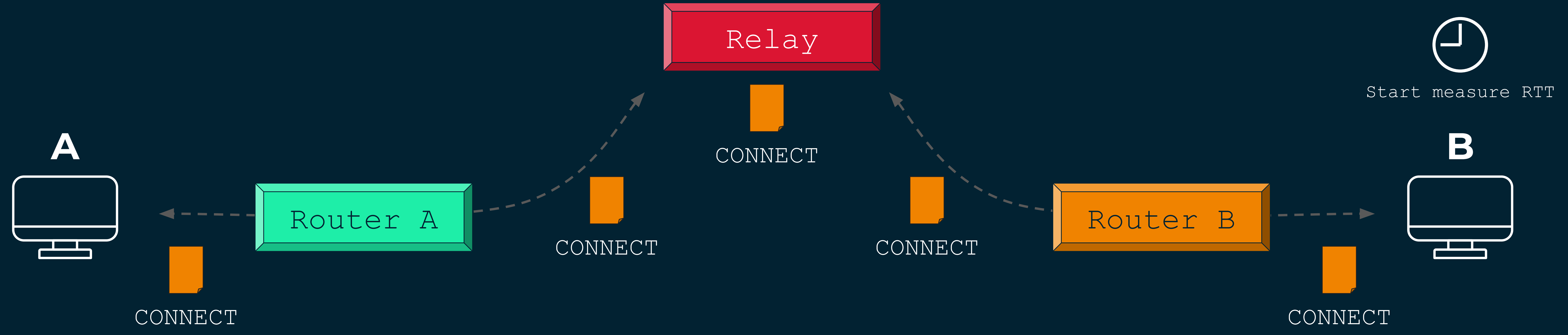
DCUtR



/<RELAY_ADDR>/p2p-circuit/<PEER_ID_B>

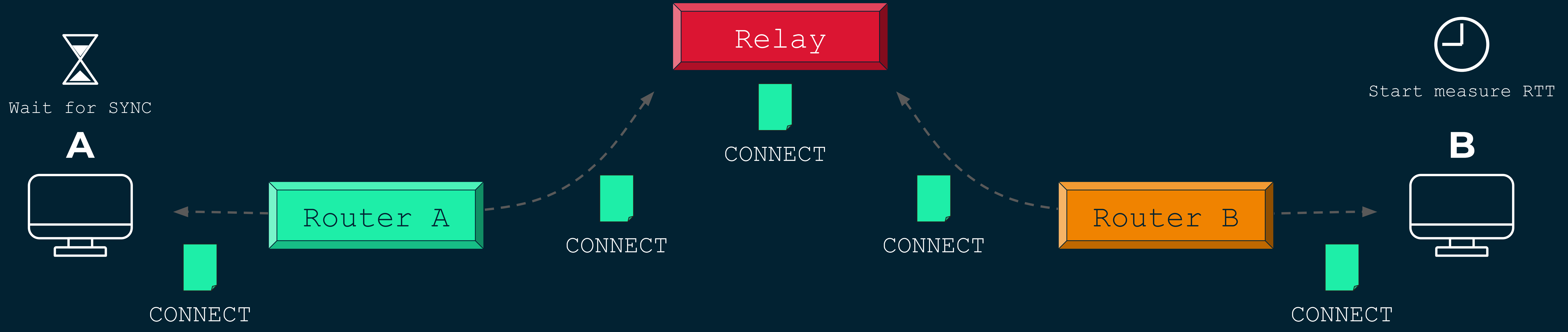
Hole Punching

DCUtR



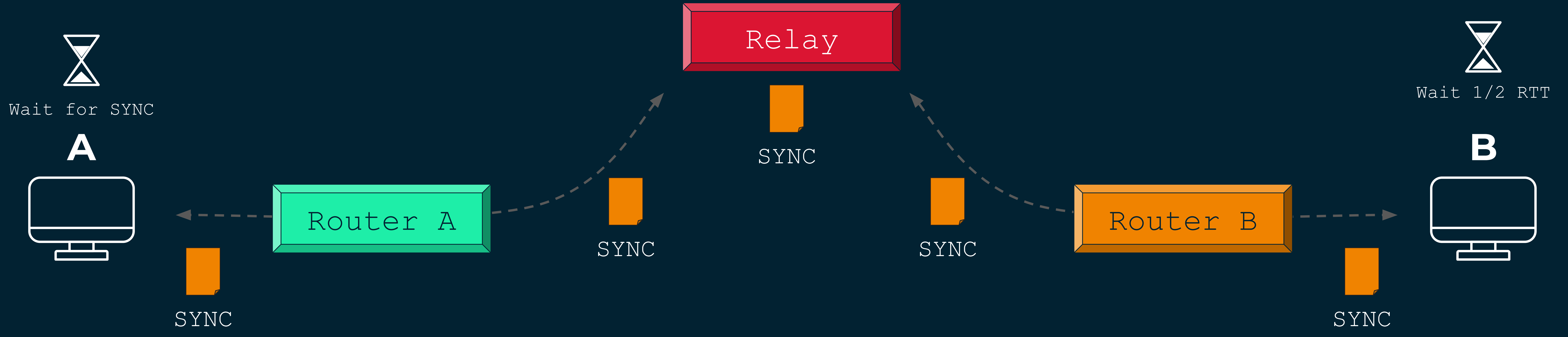
Hole Punching

DCUtR



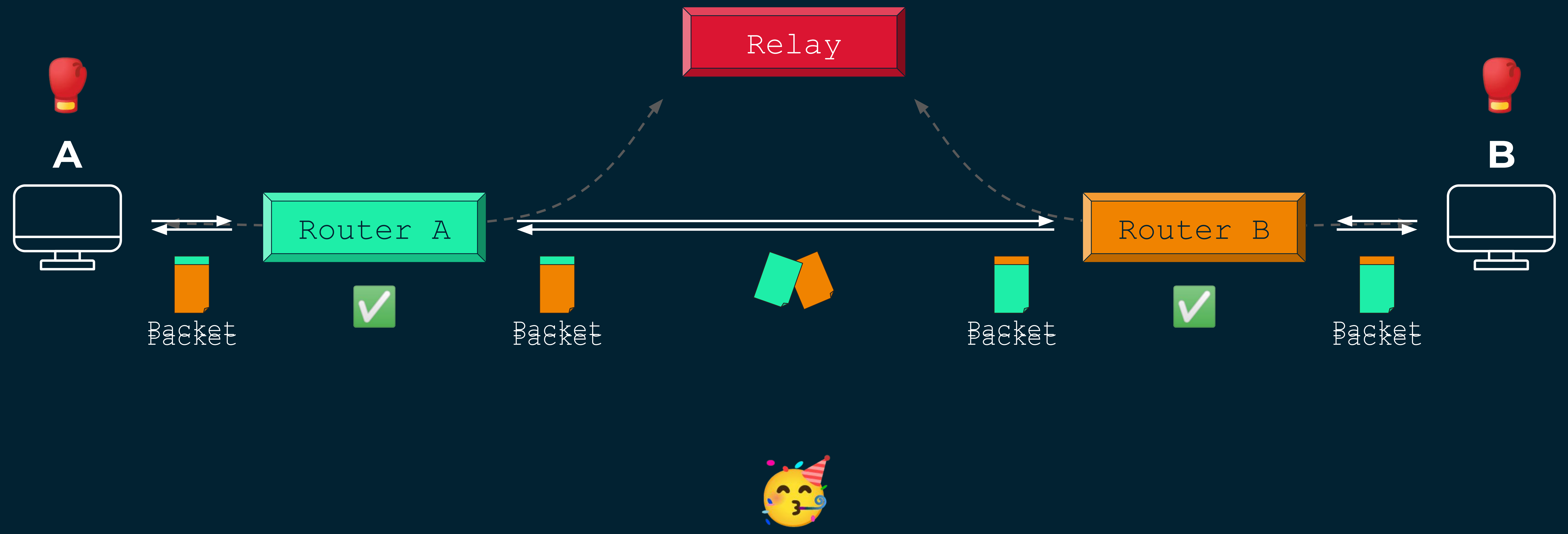
Hole Punching

DCUtR



Hole Punching

DCUtR



Timeline

2008

RFC 5128

> hole punching relies on the properties of EIM-NATs to allow appropriately designed peer-to-peer applications to "punch holes" through the NAT device(s) enroute and establish direct connectivity with each other, even when both communicating hosts lie behind NAT devices.

2021-22

Implementation in libp2p

Specification of a relay protocol and a coordination / signaling protocol. Implementation for TCP and QUIC in go-libp2p and rust-libp2p.

2022 Feb

FOSDEM 2022

Peer-to-peer hole punching without centralized infrastructure

How libp2p can traverse NATs and firewalls without coordination through central STUN and TURN servers.

2022
Summer

Role out on the IPFS network

Start with relay capabilities on all public nodes, followed by hole punching capabilities on all private nodes.

2022 Dec

Hole punching month

Measurement campaign to gain insights across networks, routers, endpoints, …

Measurements

Punchr

Architecture

Honeypot

- DHT Server
- Announces itself to the network
- Tracks inbound connections

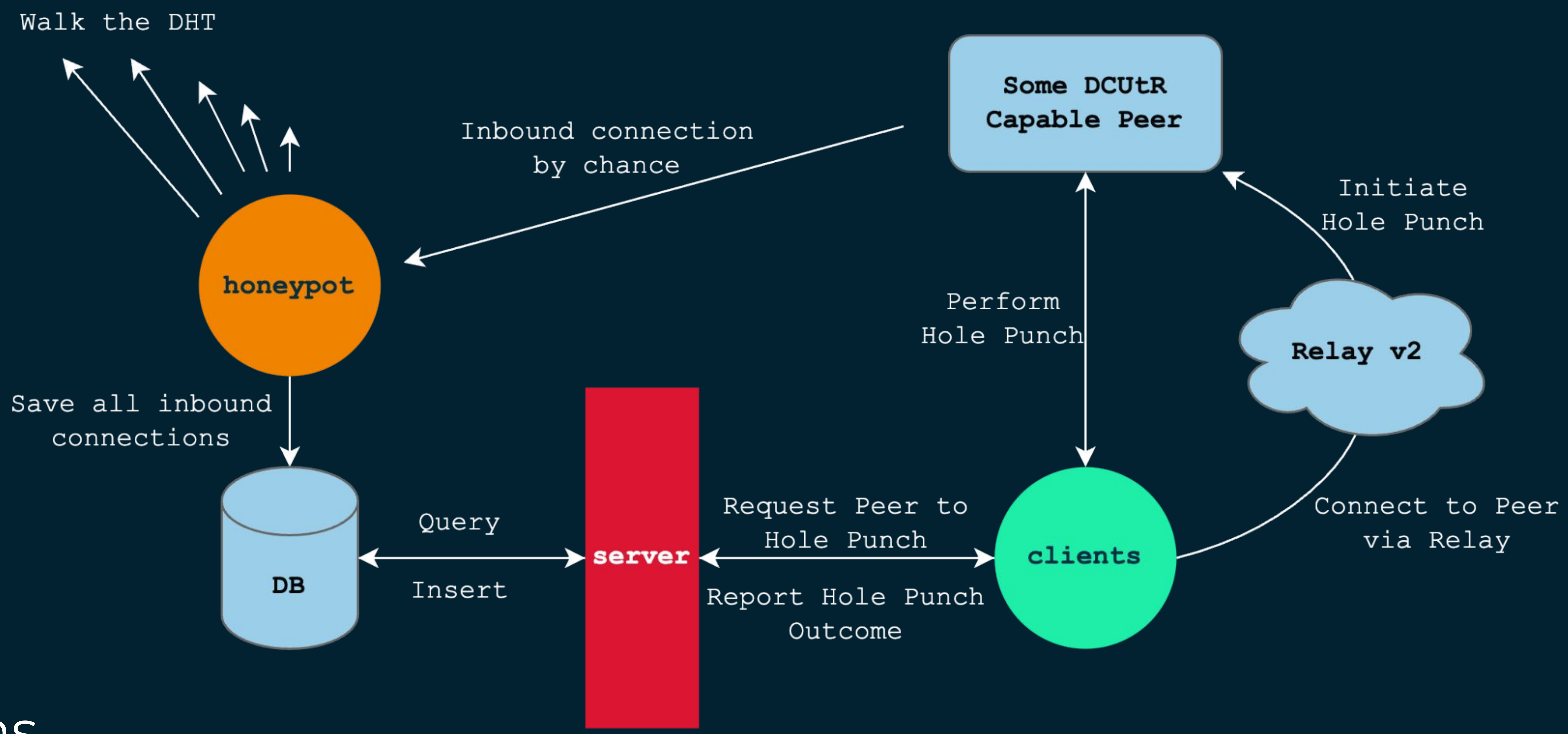
Server

- Exposes gRPC API
- Query for recently seen NAT'd DCUtR peers
- Track Results

Clients

- Rust and Go implementations
- Periodically queries server
- Reports hole punch outcome

https://github.com/libp2p/punchr


Walk the DHT
honeypot
Inbound connection by chance
Some DCUtR Capable Peer
Initiate Hole Punch
Perform Hole Punch
Relay v2
Save all inbound connections
DB
Query
Insert
server
Request Peer to Hole Punch
Report Hole Punch Outcome
clients
Connect to Peer via Relay

Measurement Results

General

Measurement Campaign

- From 2022-12-01
- To 2023-01-01 (2023-01-10)

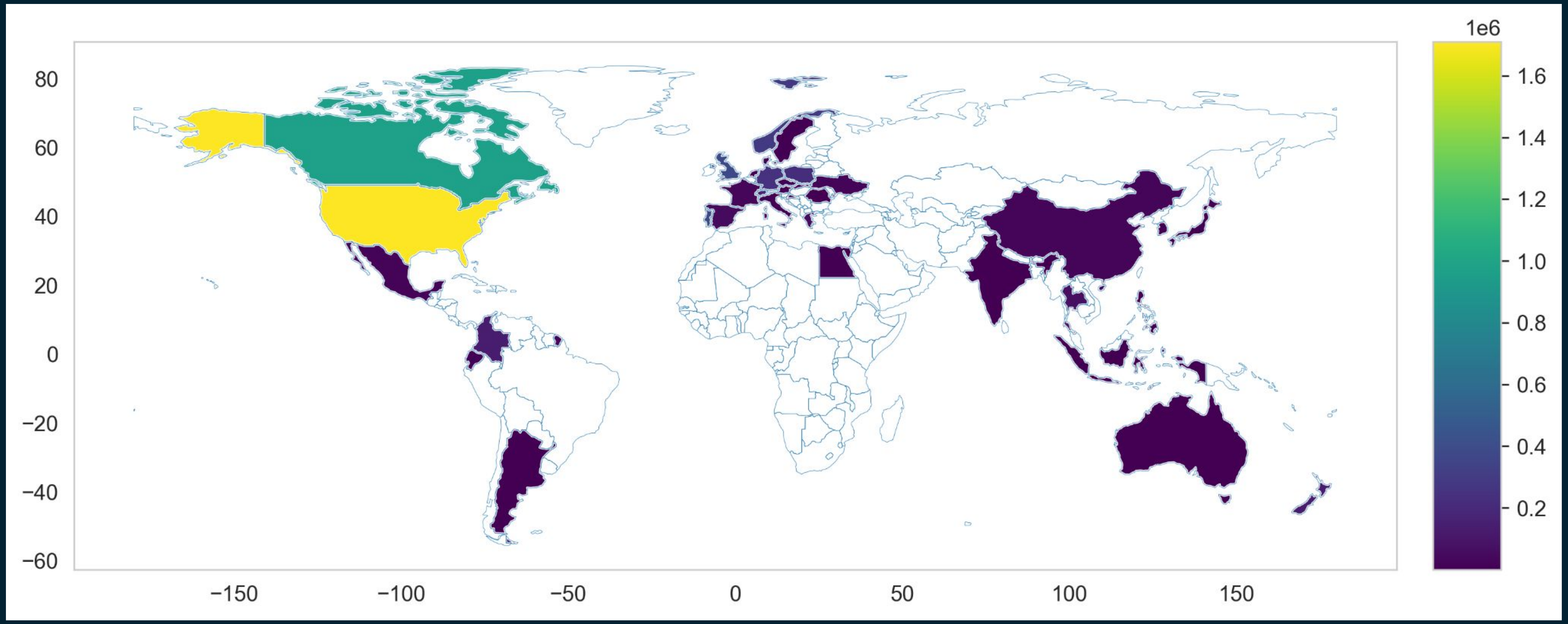
Statistics

- **>6.25M** Hole Punch Results Reported
- **154** Clients *punched* **>47k** Peers

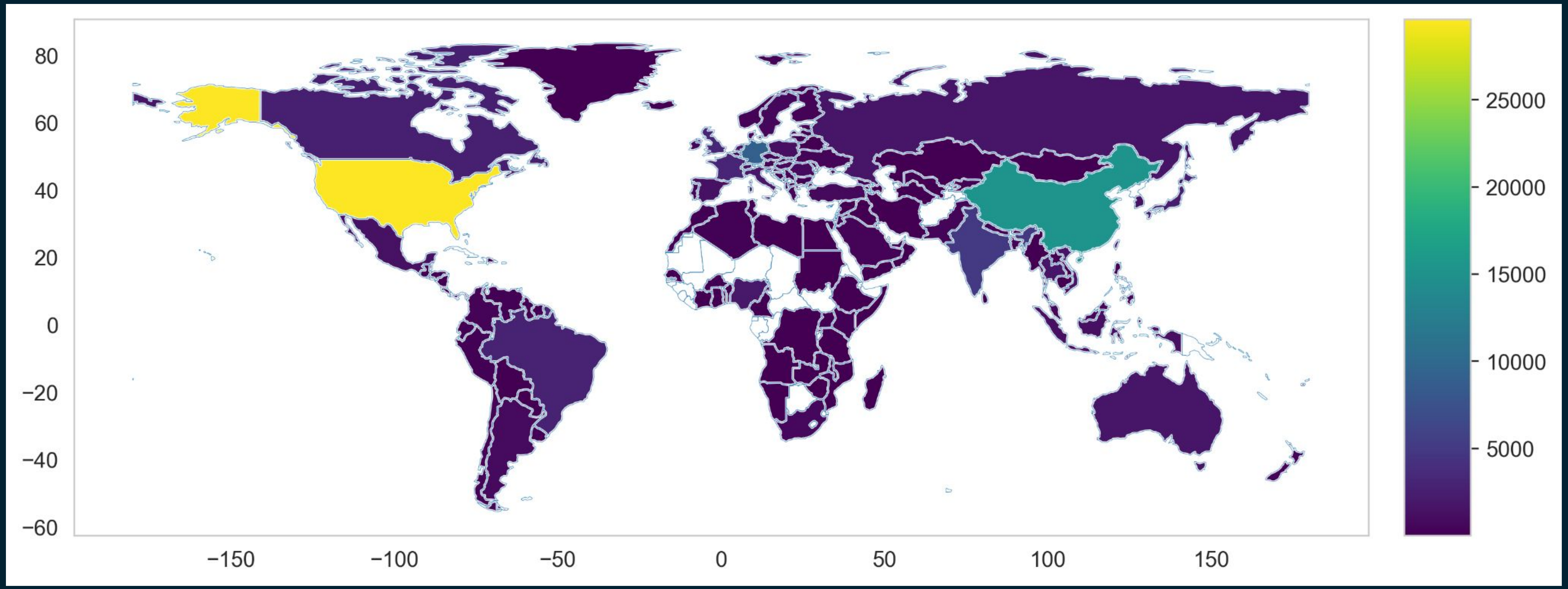
Outcomes

- NO_CONNECTION (~795k)
- NO_STREAM (~369k)
- CONNECTION_REVERSED (~711k)
- SUCCESS (~2.50M)
- FAILED (~1.88M)

Punchr Clients

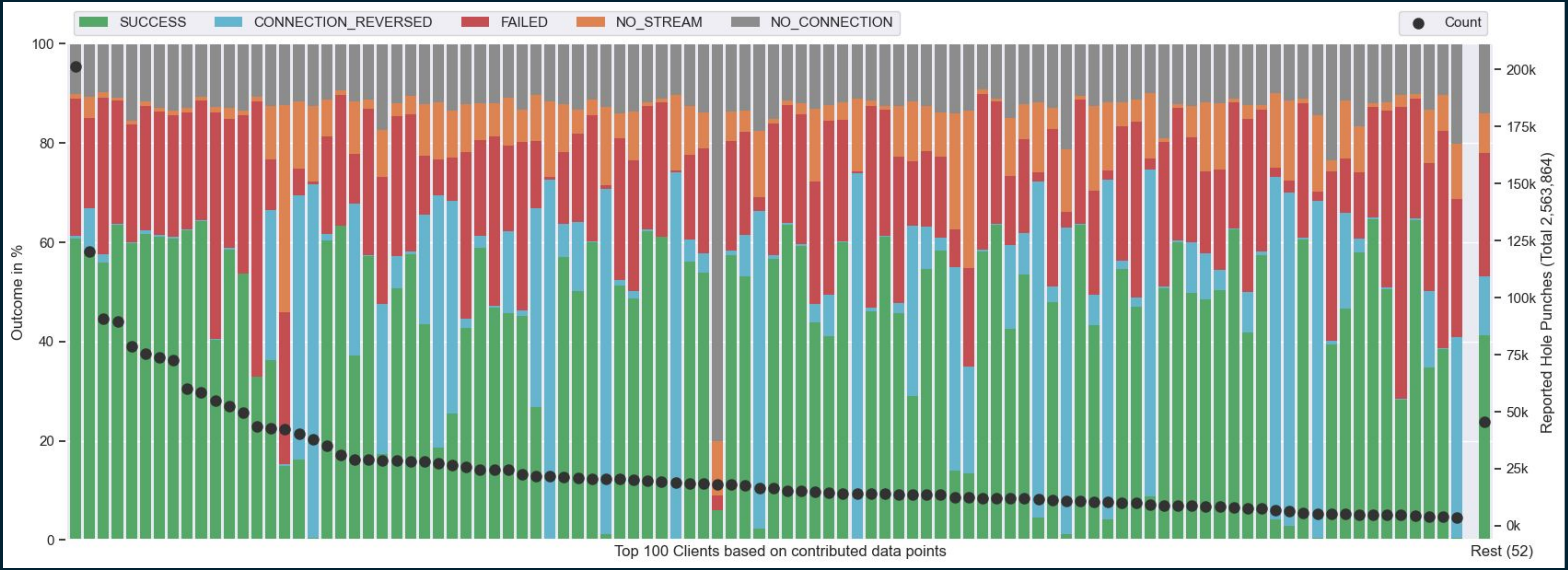


Remote Peers



Measurement Results

Client Contributions/Outcomes



Measurement Results

Network Detection

Grouping by Clients distorts the results

- Hole Punching is dependent on network setup
- A single client can be in multiple networks

How to detect individual networks?

- Group by public IP addresses/ASN
- Group by private IP addresses/ASN

But remote peers are also in different networks!

- Clients are randomly punching remote peers
- Effect applies to all clients equally
- Effect will average out

Open for other suggestions!

Example

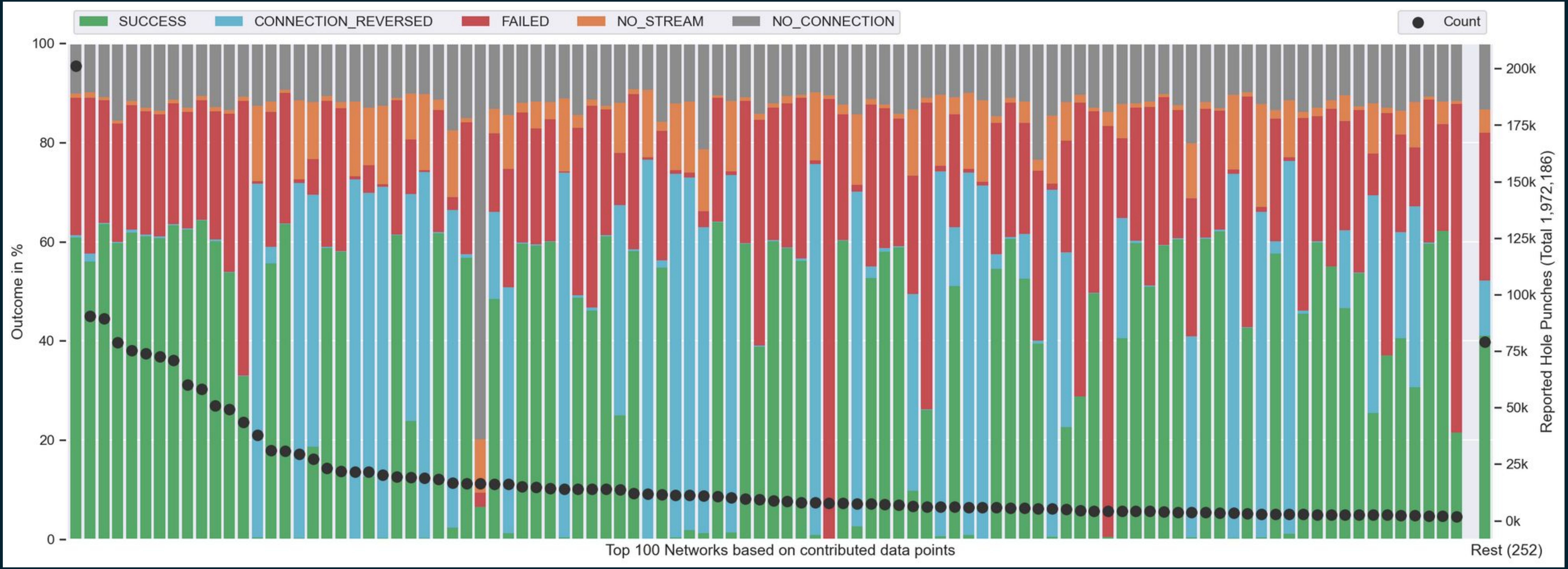
- HP1:
  - 100.100.100.100
  - 2a02:1000:9999:...
- HP2:
  - 100.100.100.100
  - 2a02:1000:7777:...
- One network

Results

- **342** unique client networks

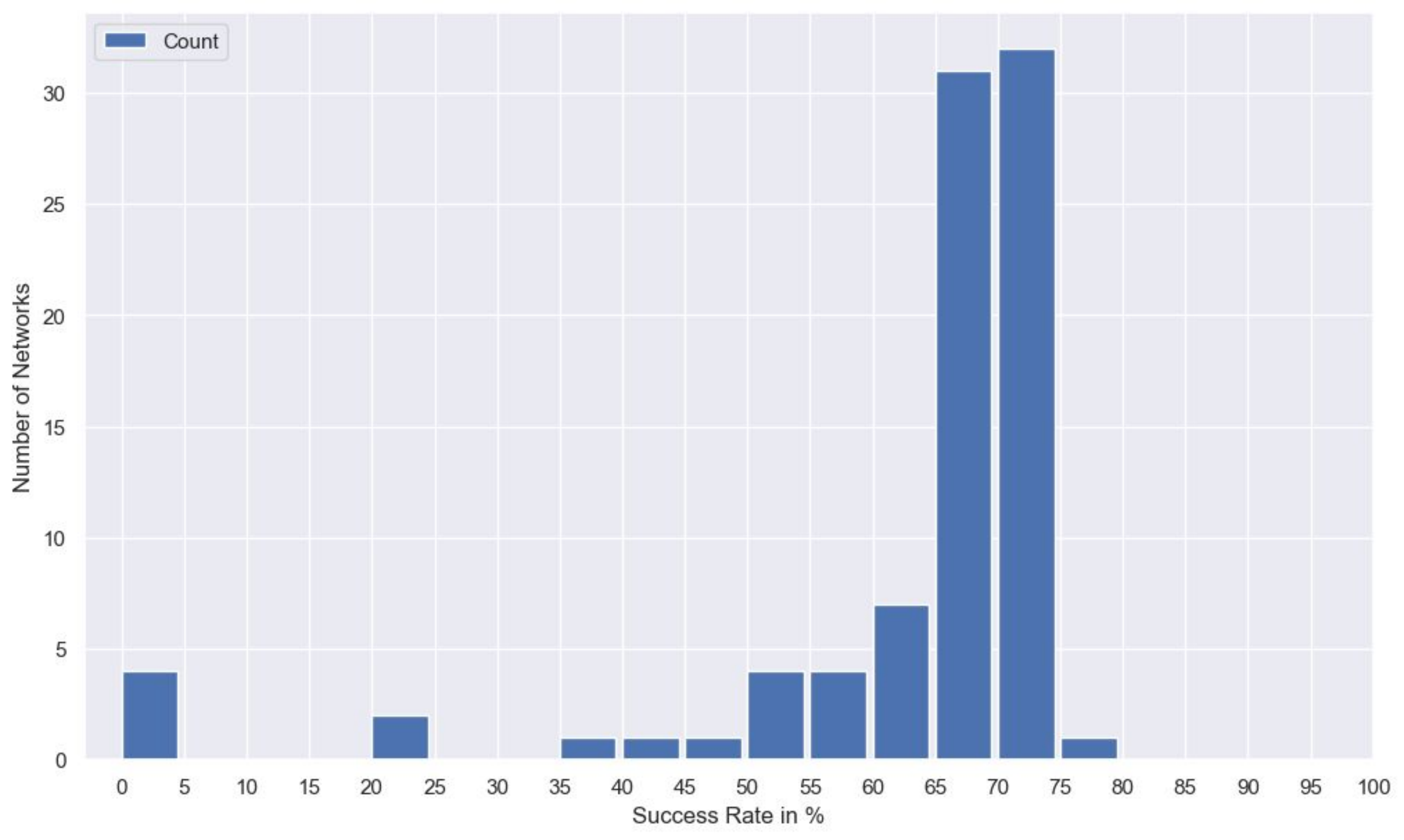
Measurement Results

Network Contributions/Outcomes



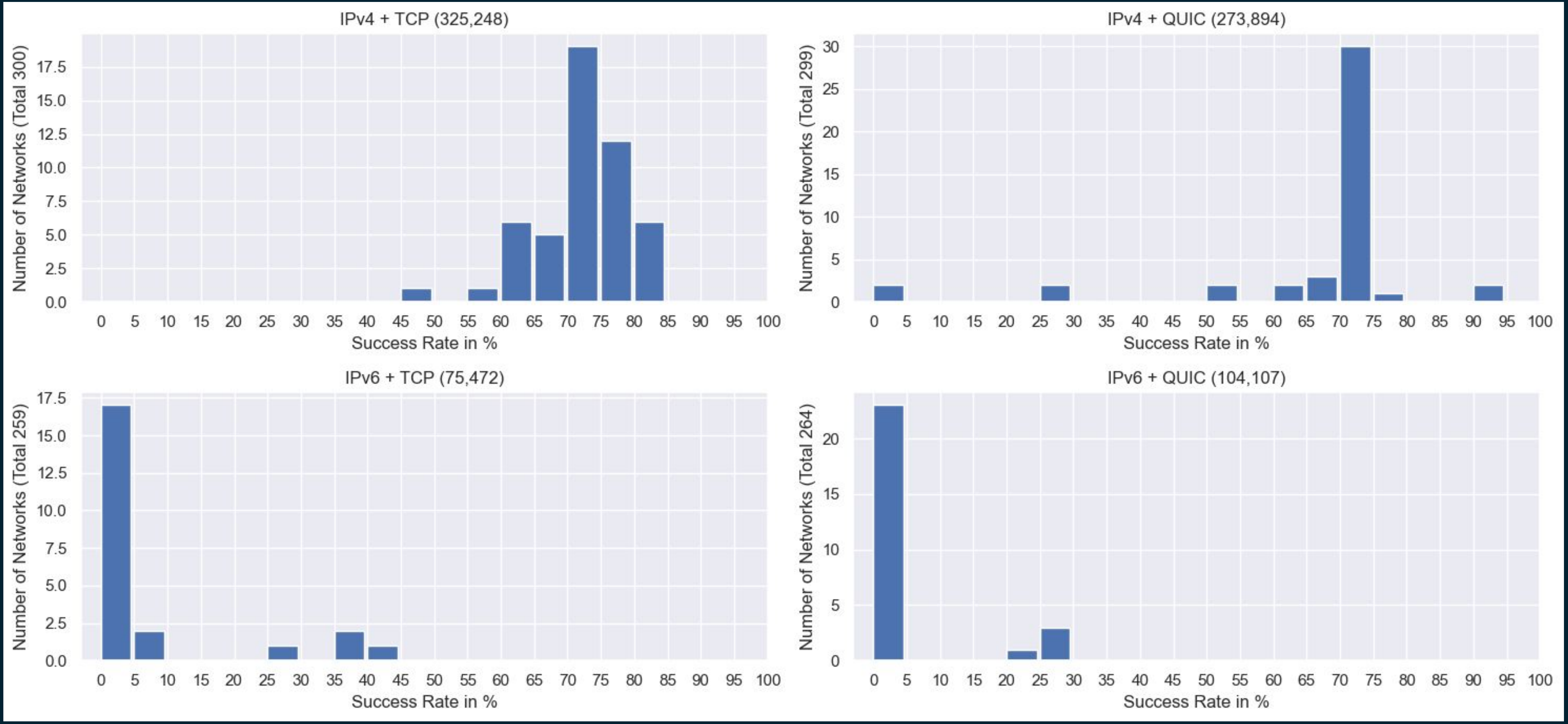
Measurement Results

Success Rate Distribution



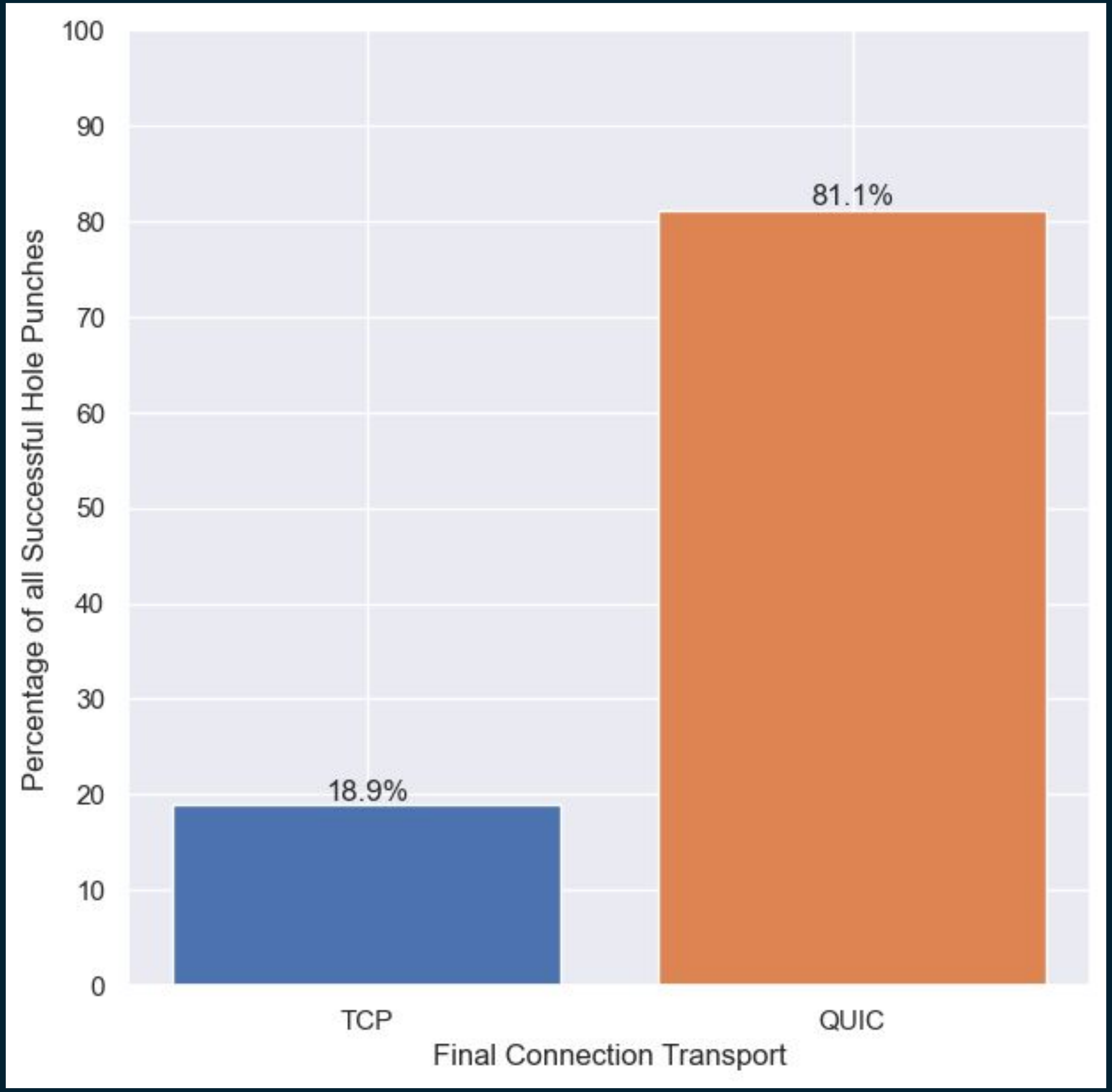
Measurement Results

IP/Transport Dependence



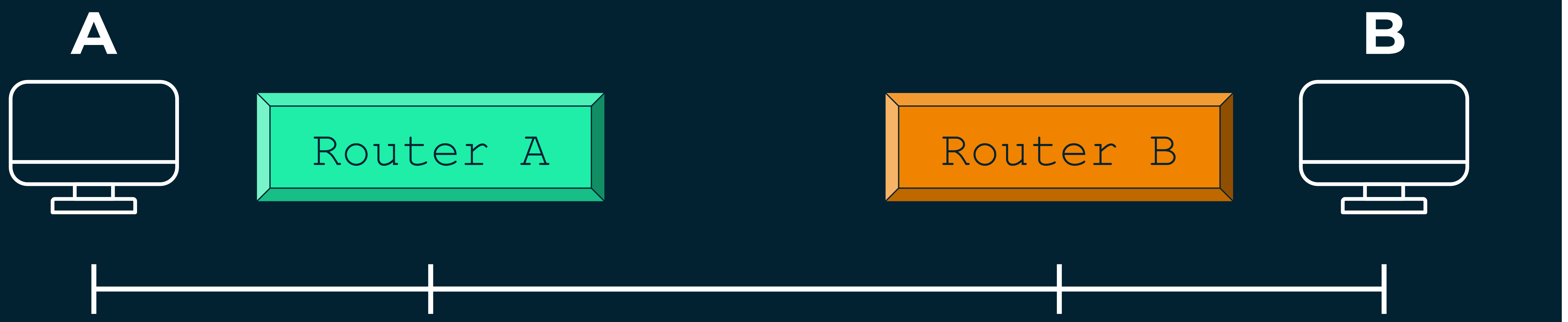
Measurement Results

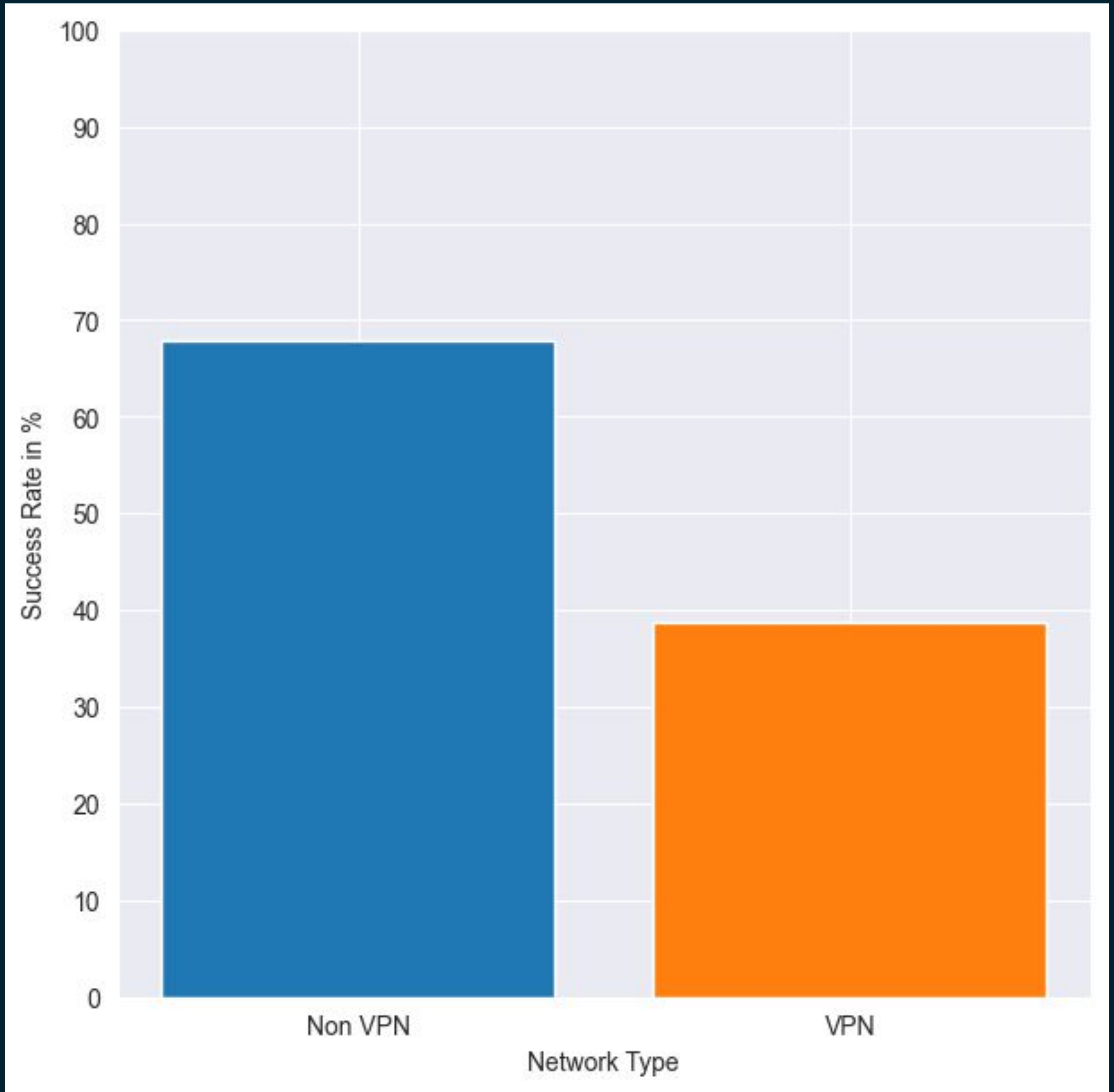
Final Connection Transport



Measurement Results

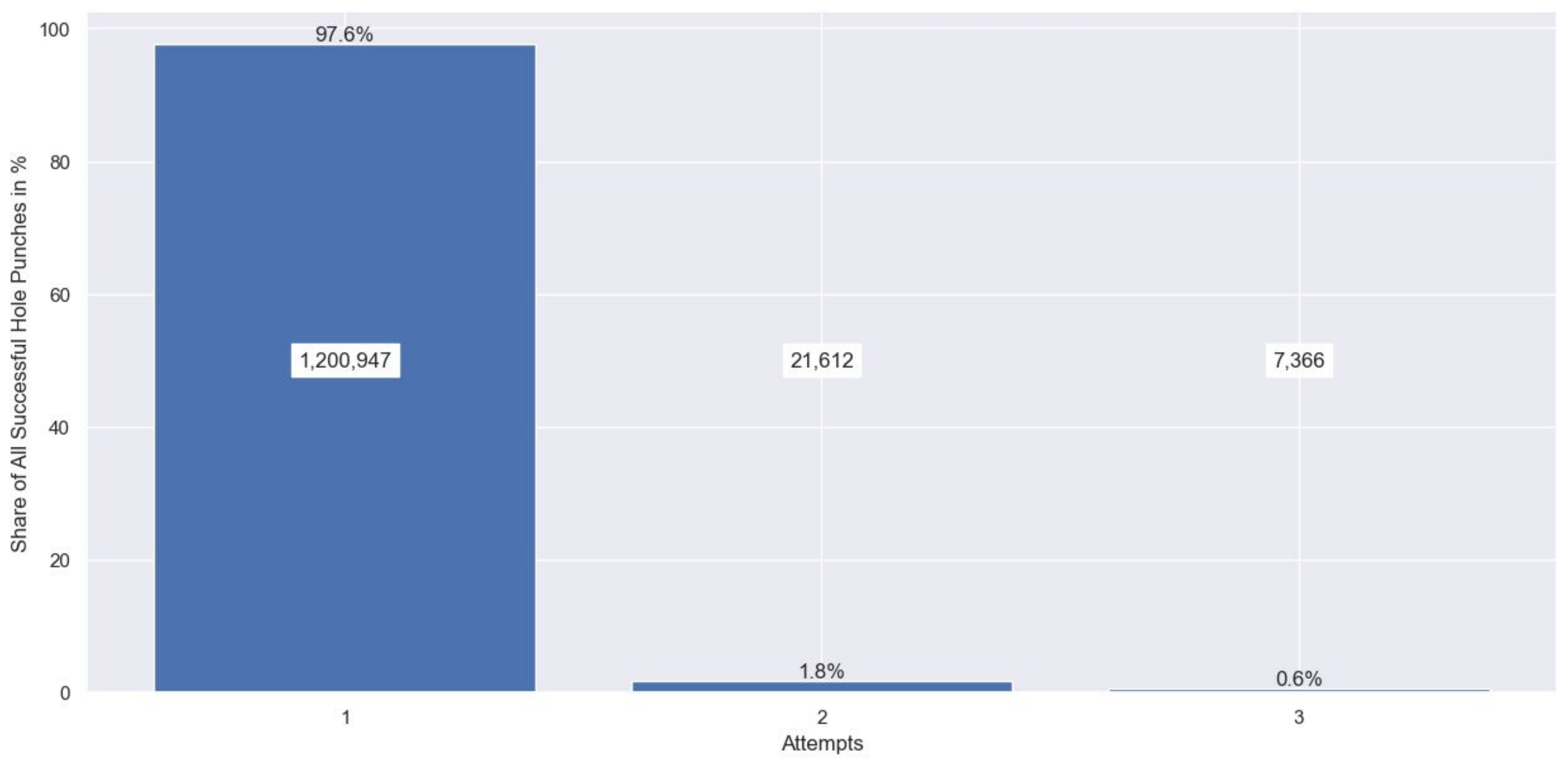
Virtual Private Network





Measurement Results

Attempts If Successful



Next Steps

Measurement Results

Next Steps

Protocol Improvements

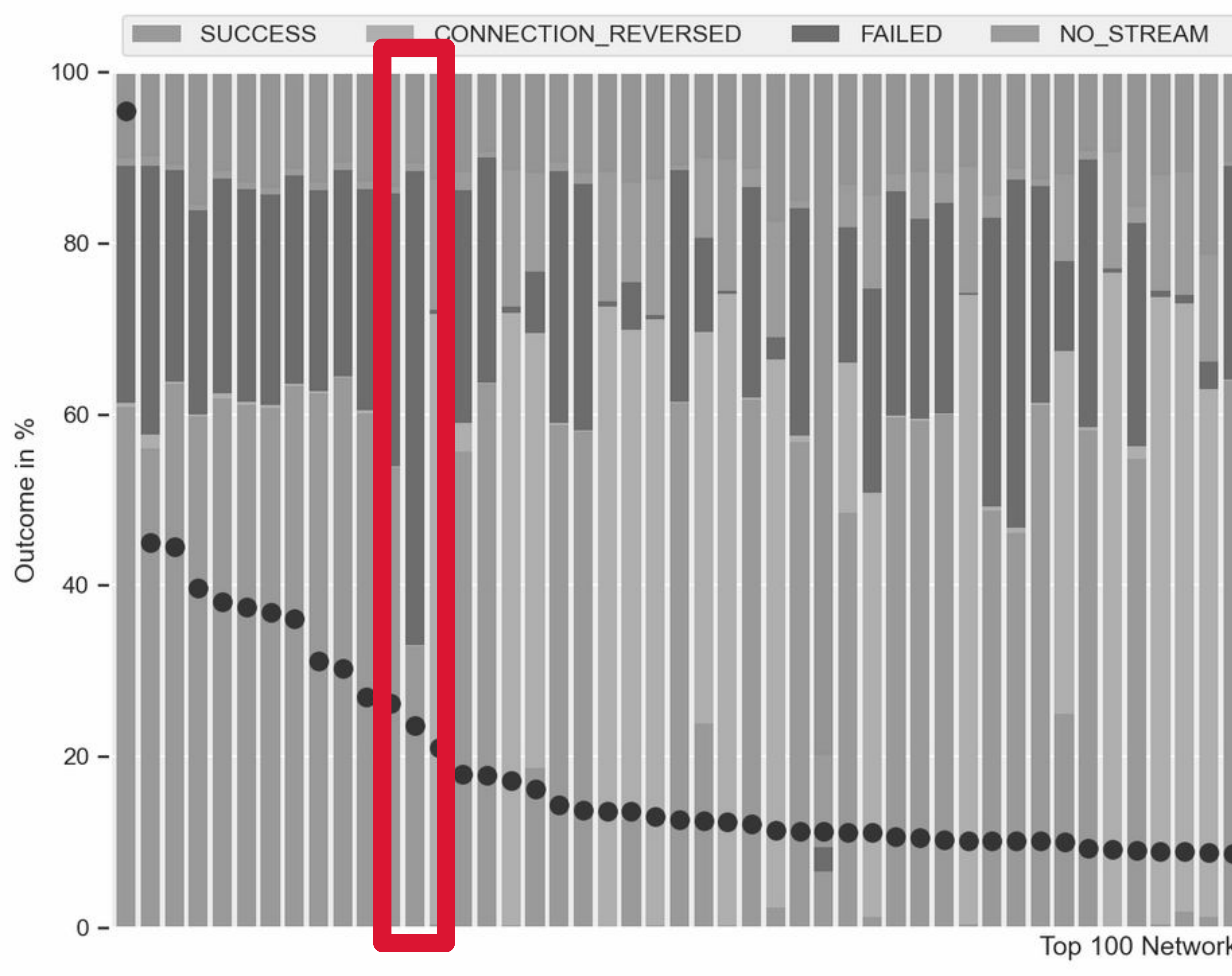
- Consider Changing Strategy on Retry
    - e.g. QUIC hole punch from both sides https://github.com/libp2p/specs/issues/487
- Measure RTT between default gateways
    - https://github.com/libp2p/specs/issues/488

Data Analysis

- Look at individual clients/networks that have low success rates
- Identify causes for hole punching problems

Academia

- Craft a follow-up publication

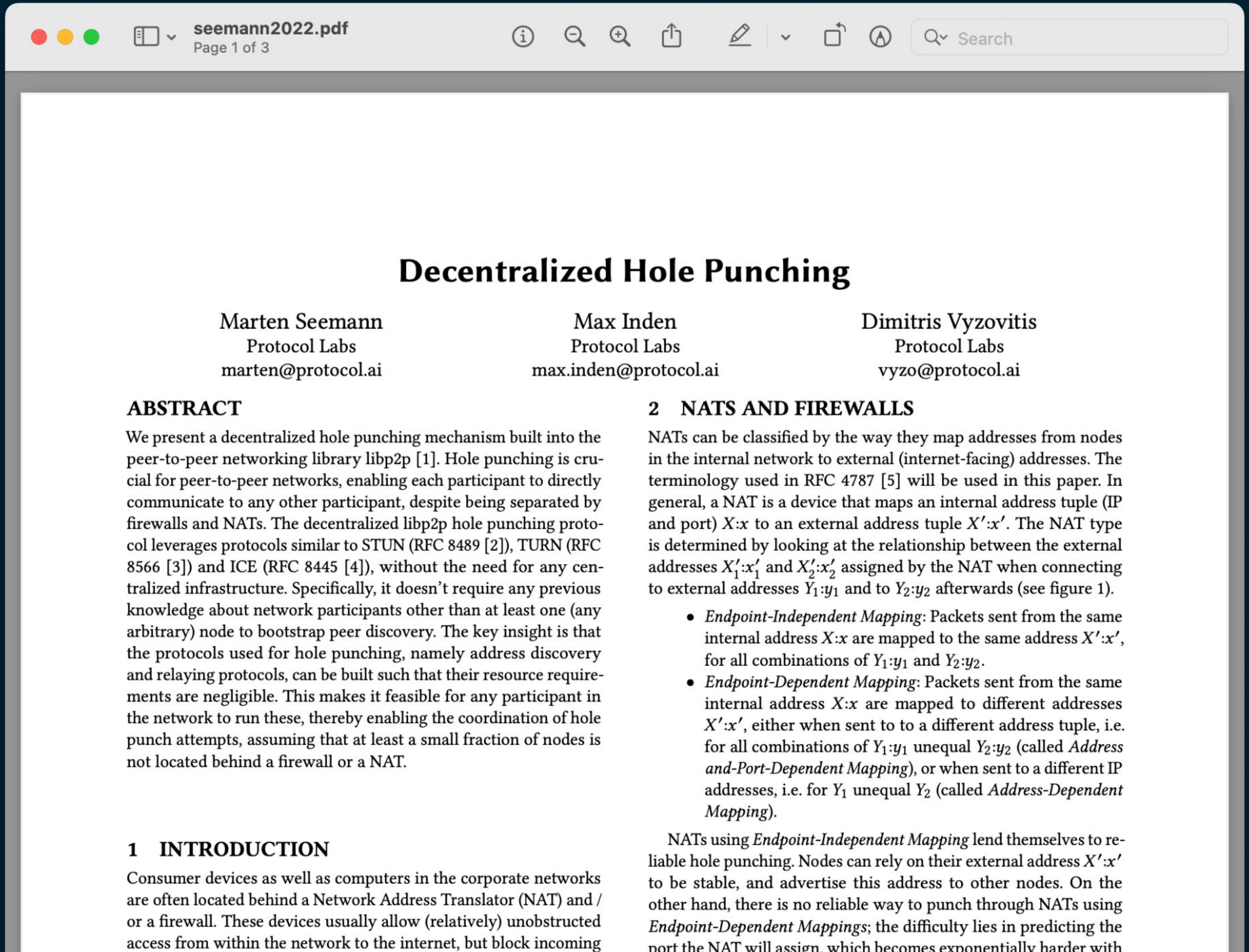


Decentralized Hole Punching Paper

"Decentralized Hole Punching." – Seemann, Marten, Max Inden, and Dimitris Vyzovitis.

2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, 2022.

QmTowSVmJTaTuwxz6ze76ifCo68GGx1HKKqJGvnYww4uBw



Decentralized Hole Punching

Marten Seemann
Protocol Labs
marten@protocol.ai

Max Inden
Protocol Labs
max.inden@protocol.ai

Dimitris Vyzovitis
Protocol Labs
vyzo@protocol.ai

ABSTRACT

We present a decentralized hole punching mechanism built into the peer-to-peer networking library libp2p [1]. Hole punching is crucial for peer-to-peer networks, enabling each participant to directly communicate to any other participant, despite being separated by firewalls and NATs. The decentralized libp2p hole punching protocol leverages protocols similar to STUN (RFC 8489 [2]), TURN (RFC 8566 [3]) and ICE (RFC 8445 [4]), without the need for any centralized infrastructure. Specifically, it doesn't require any previous knowledge about network participants other than at least one (any arbitrary) node to bootstrap peer discovery. The key insight is that the protocols used for hole punching, namely address discovery and relaying protocols, can be built such that their resource requirements are negligible. This makes it feasible for any participant in the network to run these, thereby enabling the coordination of hole punch attempts, assuming that at least a small fraction of nodes is not located behind a firewall or a NAT.

1 INTRODUCTION

Consumer devices as well as computers in the corporate networks are often located behind a Network Address Translator (NAT) and / or a firewall. These devices usually allow (relatively) unobstructed access from within the network to the internet, but block incoming

2 NATS AND FIREWALLS

NATs can be classified by the way they map addresses from nodes in the internal network to external (internet-facing) addresses. The terminology used in RFC 4787 [5] will be used in this paper. In general, a NAT is a device that maps an internal address tuple (IP and port) $X{:}x$ to an external address tuple $X'{:}x'$. The NAT type is determined by looking at the relationship between the external addresses $X_1'{:}x_1'$ and $X_2'{:}x_2'$ assigned by the NAT when connecting to external addresses $Y_1{:}y_1$ and to $Y_2{:}y_2$ afterwards (see figure 1).

- *Endpoint-Independent Mapping*: Packets sent from the same internal address $X{:}x$ are mapped to the same address $X'{:}x'$, for all combinations of $Y_1{:}y_1$ and $Y_2{:}y_2$.
- *Endpoint-Dependent Mapping*: Packets sent from the same internal address $X{:}x$ are mapped to different addresses $X'{:}x'$, either when sent to to a different address tuple, i.e. for all combinations of $Y_1{:}y_1$ unequal $Y_2{:}y_2$ (called *Address and-Port-Dependent Mapping*), or when sent to a different IP addresses, i.e. for Y_1 unequal Y_2 (called *Address-Dependent Mapping*).

NATs using *Endpoint-Independent Mapping* lend themselves to reliable hole punching. Nodes can rely on their external address $X'{:}x'$ to be stable, and advertise this address to other nodes. On the other hand, there is no reliable way to punch through NATs using *Endpoint-Dependent Mappings*; the difficulty lies in predicting the port the NAT will assign, which becomes exponentially harder with

Get involved!

- Talk to us here at the venue
- Documentation - docs.libp2p.io
- Forum - discuss.libp2p.io
- Specification & Roadmap - github.com/libp2p/specs
- Implementations - github.com/libp2p/<LANGUAGE>-libp2p
- Join the community call

Github: @mxinden

Twitter: @mxinden

Web: https://max-inden.de

Email: mail@max-inden.de

Github: @dennis-tra

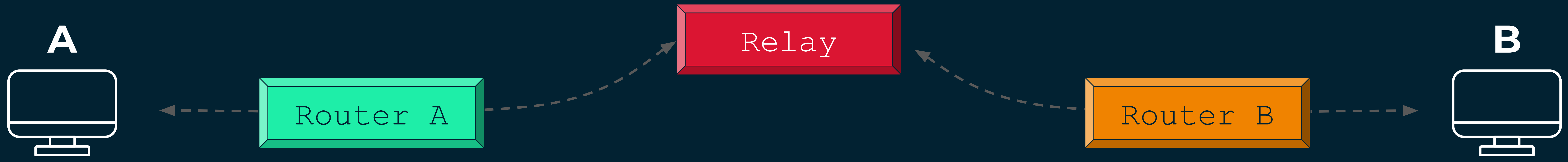
Twitter: @dtrautwein_eu

Web: https://dtrautwein.eu

Email: dennis@protocol.ai

Measurement Results

Round Trip Time Dependence



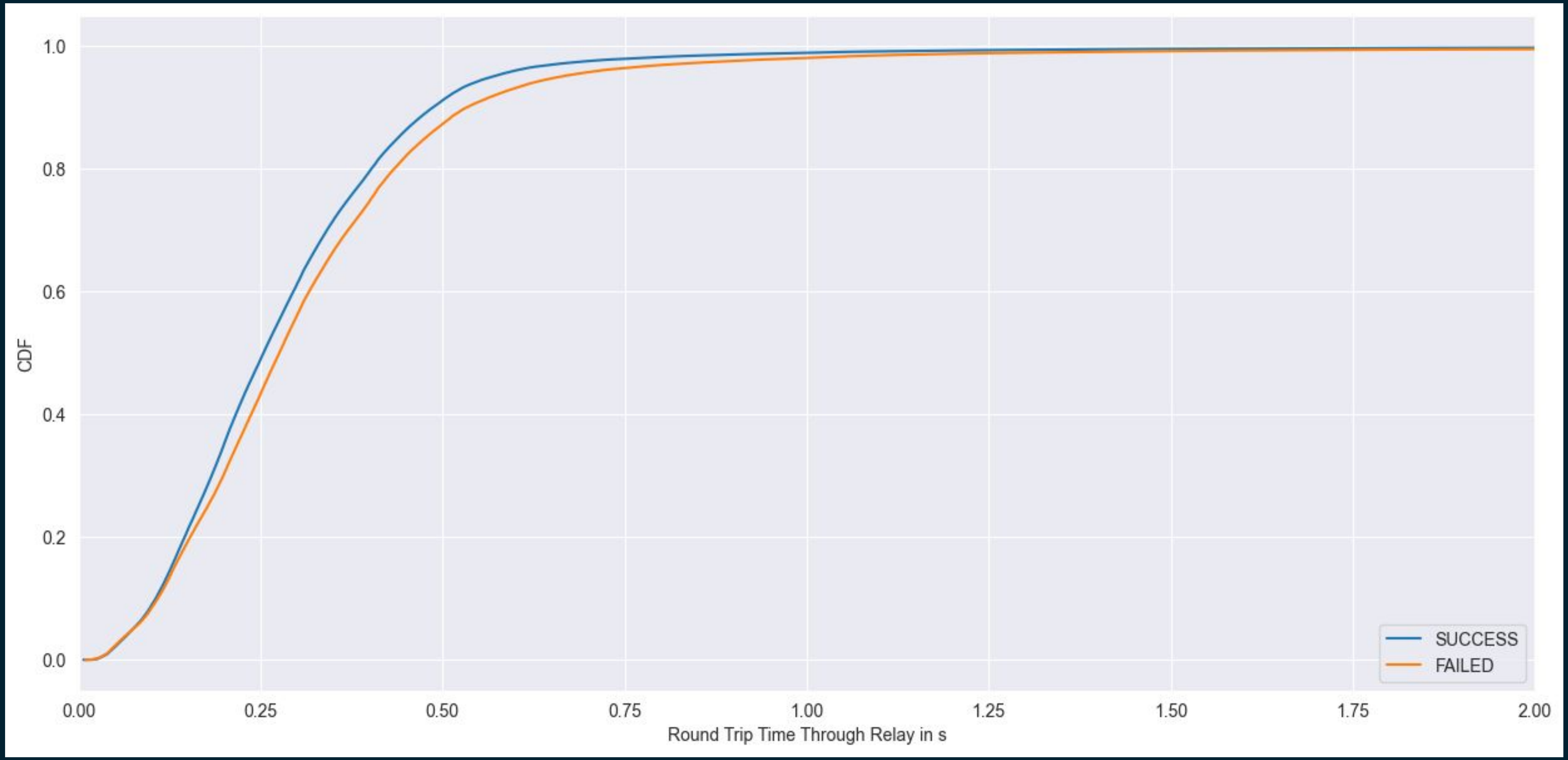
A

Router A

Relay

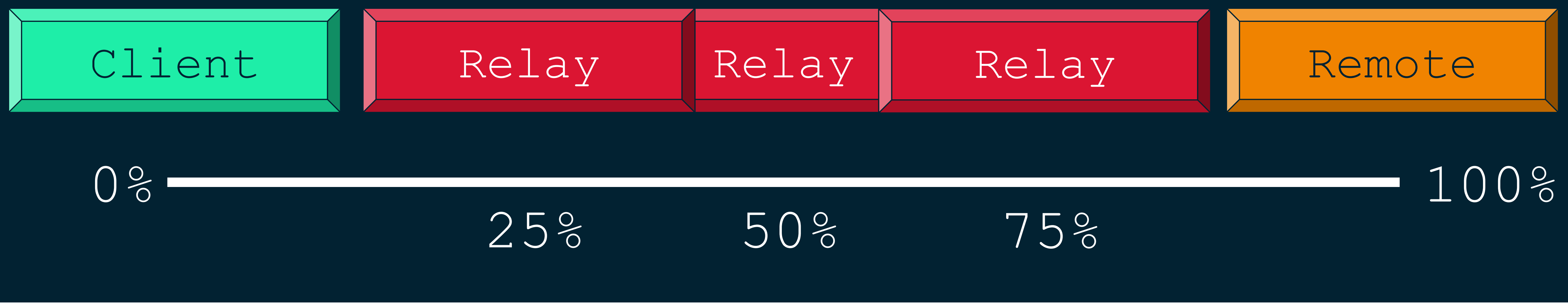
Router B

B



Measurement Results

Relay Location Dependence



Client

Relay

Relay

Relay

Remote

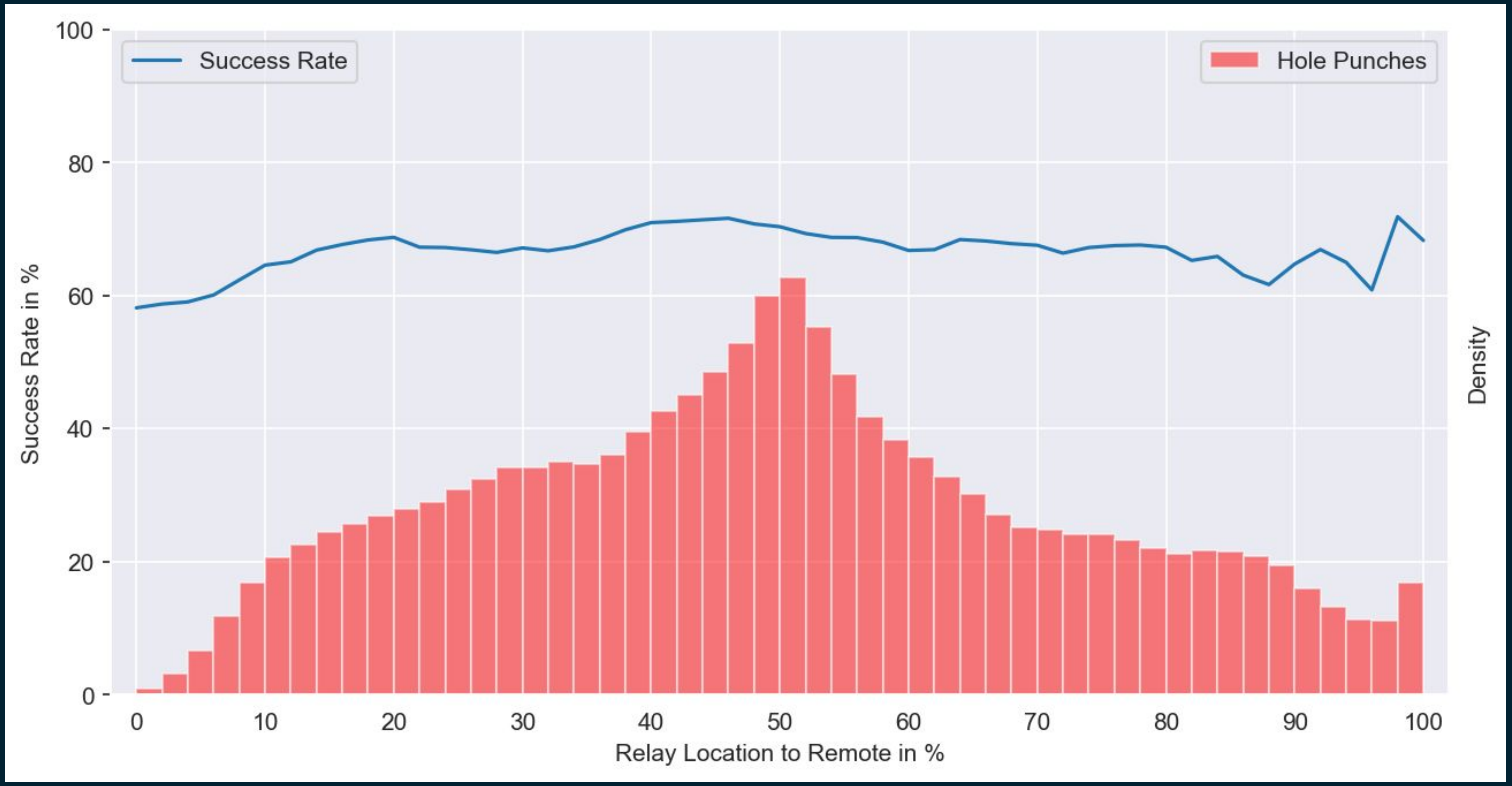
0%

25%

50%

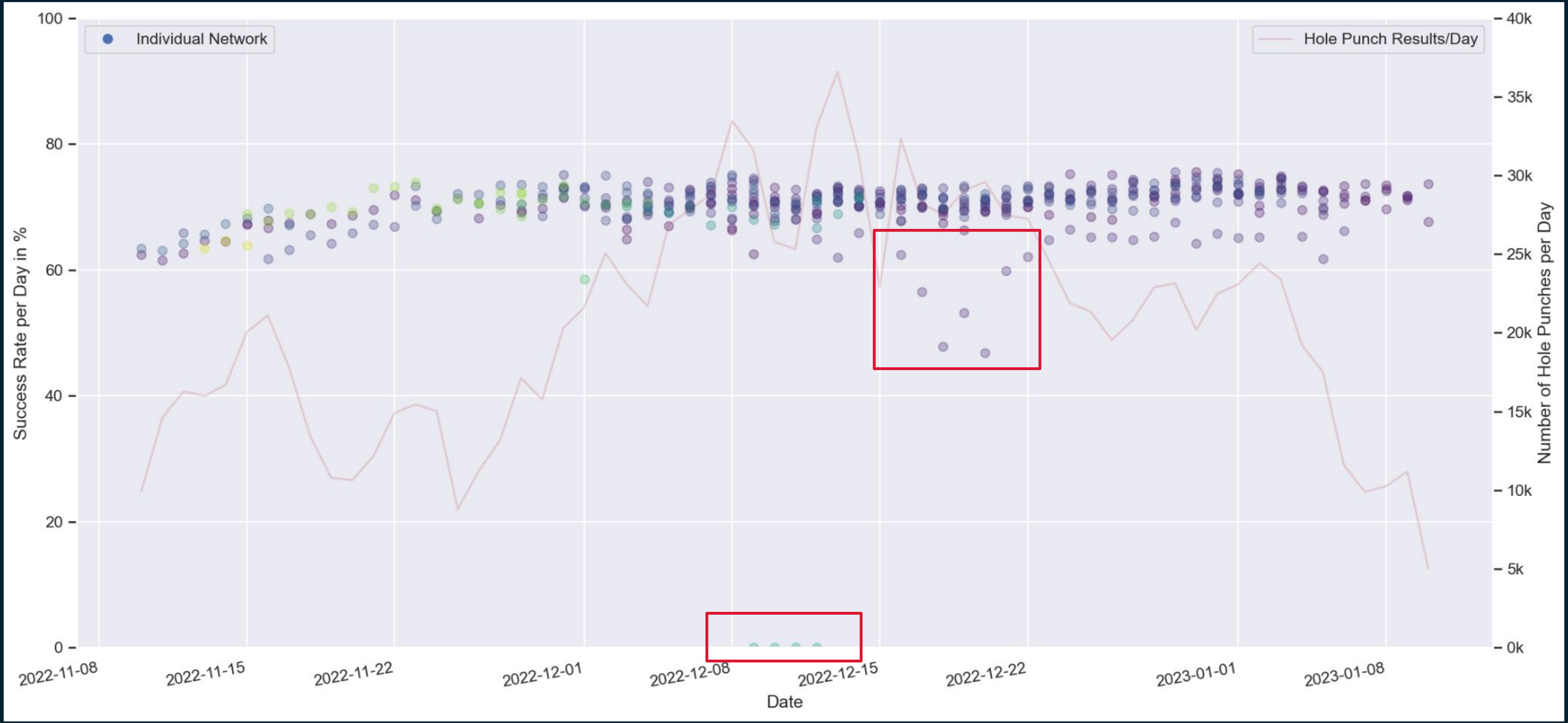
75%

100%



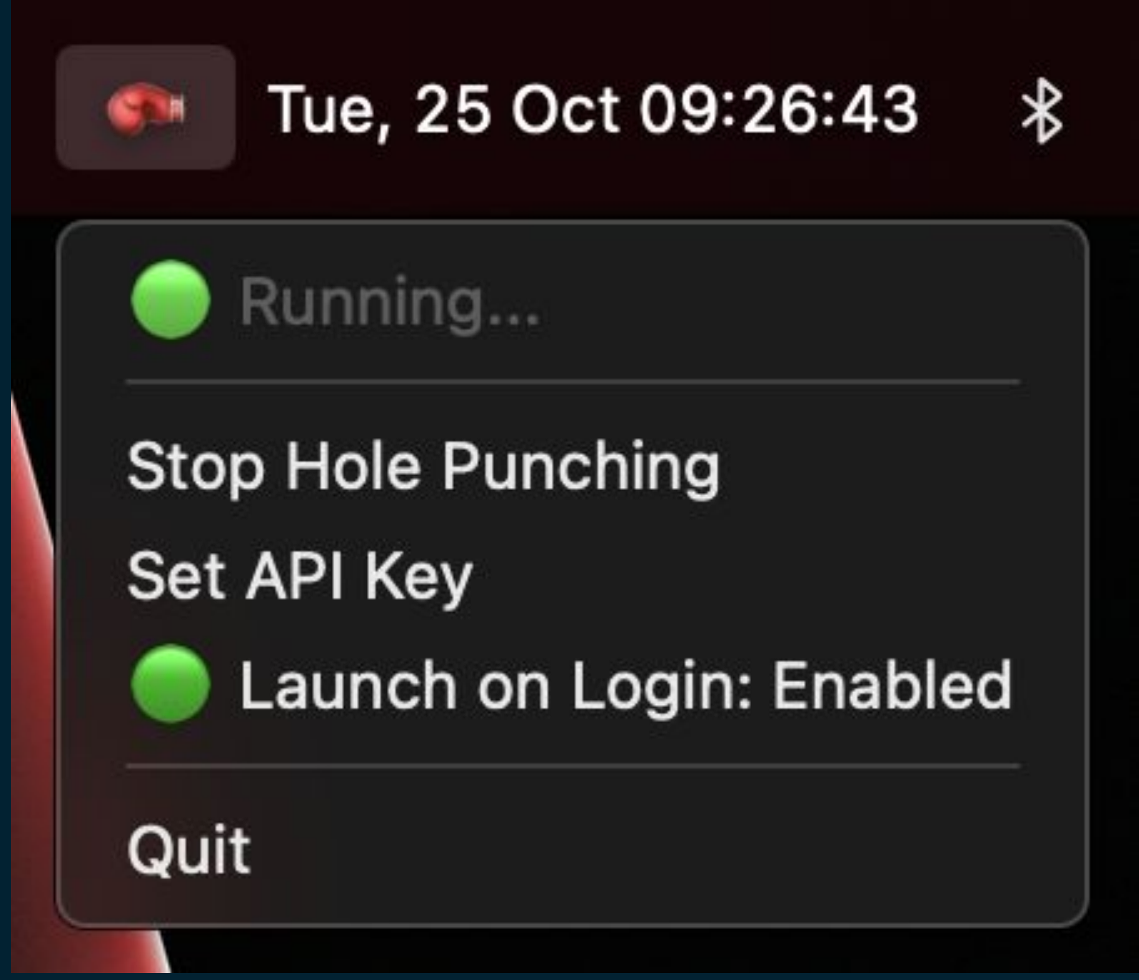
Measurement Results

Success Rate over Time



Punchr

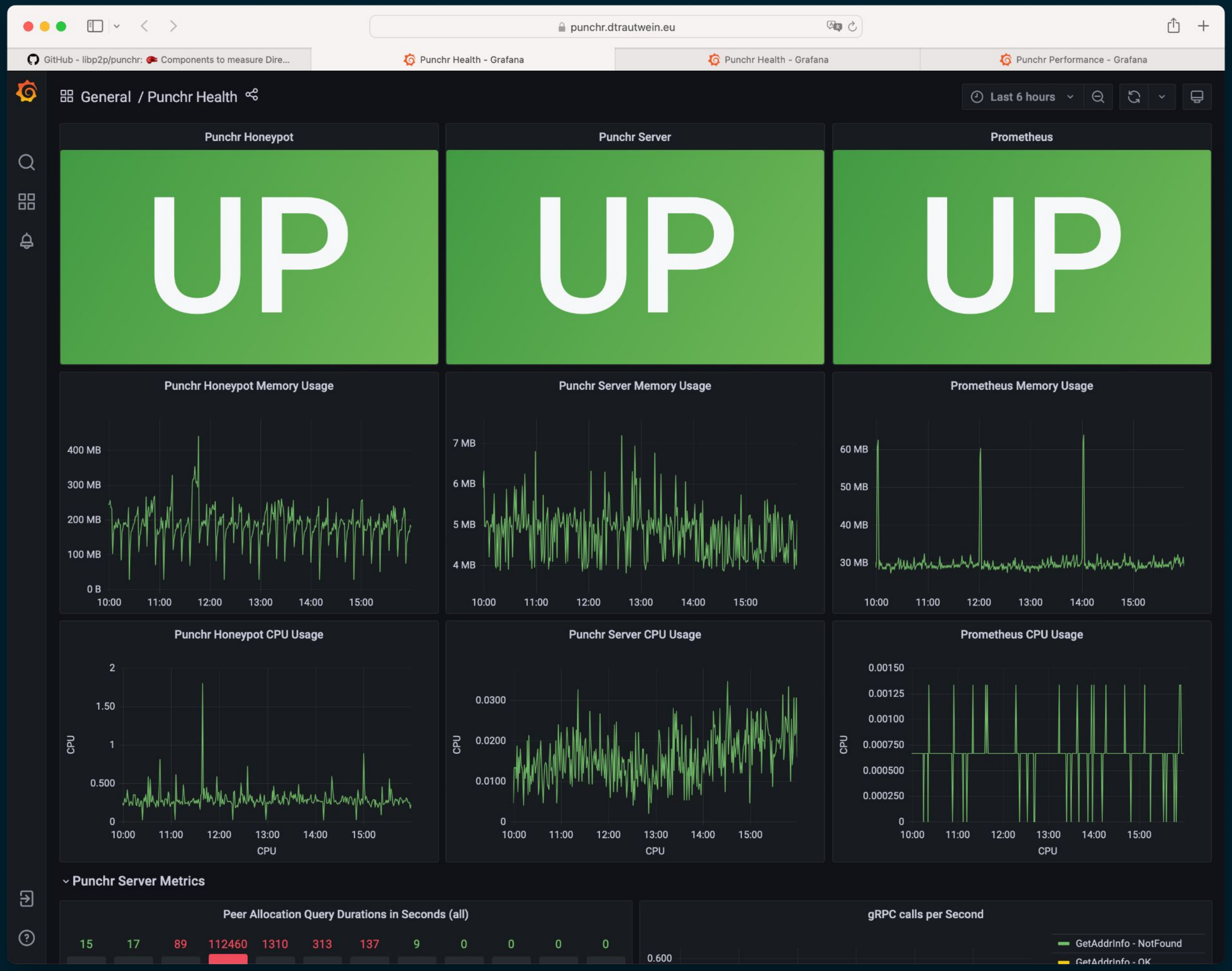
Monitoring

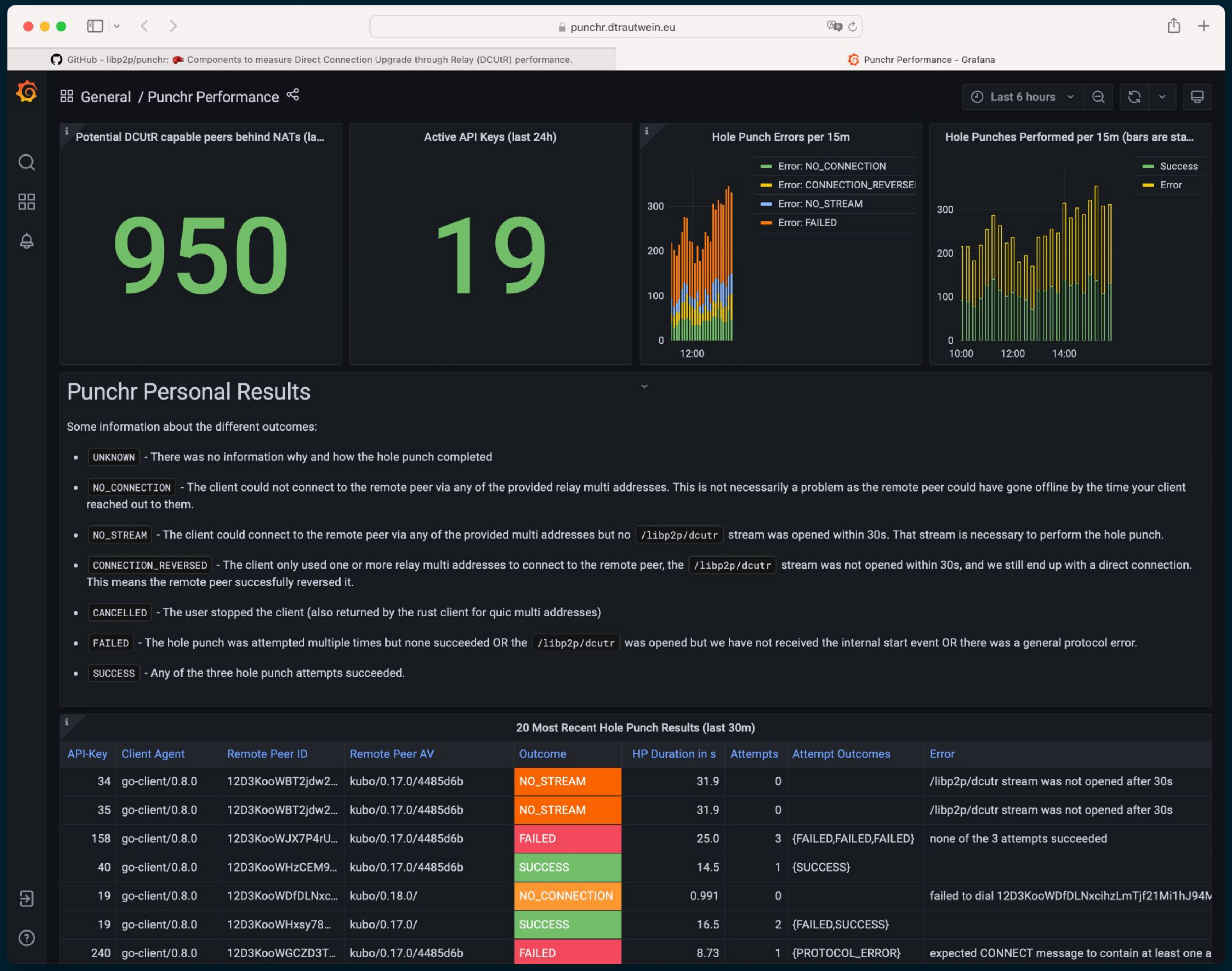


Dashboards

- Health
- Performance

https://punchr.dtrautwein.eu/grafana/




punchr.dtrautwein.eu
GitHub - libp2p/punchr: Components to measure Direct Connection Upgrade through Relay (DCUtR) performance.
Punchr Performance - Grafana
General / Punchr Performance
Last 6 hours
Potential DCUtR capable peers behind NATs (la...
950
Active API Keys (last 24h)
19
Hole Punch Errors per 15m
Error: NO_CONNECTION
Error: CONNECTION_REVERSE
Error: NO_STREAM
Error: FAILED
Hole Punches Performed per 15m (bars are sta...
Success
Error

Punchr Personal Results

Some information about the different outcomes:

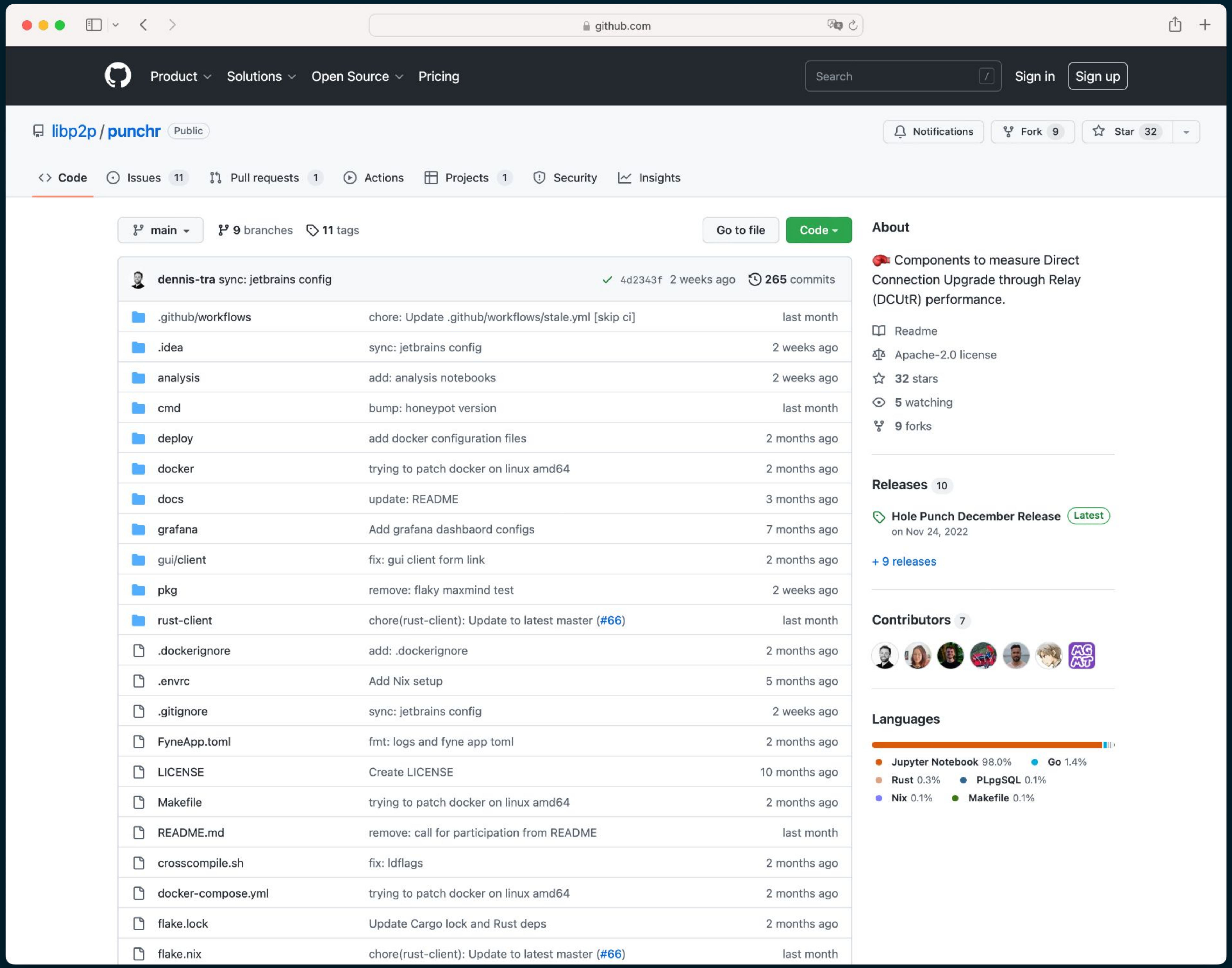
- `UNKNOWN` - There was no information why and how the hole punch completed
- `NO_CONNECTION` - The client could not connect to the remote peer via any of the provided relay multi addresses. This is not necessarily a problem as the remote peer could have gone offline by the time your client reached out to them.
- `NO_STREAM` - The client could connect to the remote peer via any of the provided multi addresses but no `/libp2p/dcutr` stream was opened within 30s. That stream is necessary to perform the hole punch.
- `CONNECTION_REVERSED` - The client only used one or more relay multi addresses to connect to the remote peer, the `/libp2p/dcutr` stream was not opened within 30s, and we still end up with a direct connection. This means the remote peer succesfully reversed it.
- `CANCELLED` - The user stopped the client (also returned by the rust client for quic multi addresses)
- `FAILED` - The hole punch was attempted multiple times but none succeeded OR the `/libp2p/dcutr` was opened but we have not received the internal start event OR there was a general protocol error.
- `SUCCESS` - Any of the three hole punch attempts succeeded.

20 Most Recent Hole Punch Results (last 30m)

API-Key	Client Agent	Remote Peer ID	Remote Peer AV	Outcome	HP Duration in s	Attempts	Attempt Outcomes	Error
34	go-client/0.8.0	12D3KooWBT2jdw2...	kubo/0.17.0/4485d6b	NO_STREAM	31.9	0		/libp2p/dcutr stream was not opened after 30s
35	go-client/0.8.0	12D3KooWBT2jdw2...	kubo/0.17.0/4485d6b	NO_STREAM	31.9	0		/libp2p/dcutr stream was not opened after 30s
158	go-client/0.8.0	12D3KooWJX7P4rU...	kubo/0.17.0/4485d6b	FAILED	25.0	3	{FAILED,FAILED,FAILED}	none of the 3 attempts succeeded
40	go-client/0.8.0	12D3KooWHzCEM9...	kubo/0.17.0/4485d6b	SUCCESS	14.5	1	{SUCCESS}	
19	go-client/0.8.0	12D3KooWDfDLNxc...	kubo/0.18.0/	NO_CONNECTION	0.991	0		failed to dial 12D3KooWDfDLNxcihzLmTjf21Mi1hJ94M
19	go-client/0.8.0	12D3KooWHxsy78...	kubo/0.17.0/	SUCCESS	16.5	2	{FAILED,SUCCESS}	
240	go-client/0.8.0	12D3KooWGCZD3T...	kubo/0.17.0/4485d6b	FAILED	8.73	1	{PROTOCOL_ERROR}	expected CONNECT message to contain at least one a

Punchr

Repository



libp2p / punchr Public

9 branches 11 tags

dennis-tra sync: jetbrains config ✓ 4d2343f 2 weeks ago 265 commits

.github/workflows	chore: Update .github/workflows/stale.yml [skip ci]	last month
.idea	sync: jetbrains config	2 weeks ago
analysis	add: analysis notebooks	2 weeks ago
cmd	bump: honeypot version	last month
deploy	add docker configuration files	2 months ago
docker	trying to patch docker on linux amd64	2 months ago
docs	update: README	3 months ago
grafana	Add grafana dashbaord configs	7 months ago
gui/client	fix: gui client form link	2 months ago
pkg	remove: flaky maxmind test	2 weeks ago
rust-client	chore(rust-client): Update to latest master (#66)	last month
.dockerignore	add: .dockerignore	2 months ago
.envrc	Add Nix setup	5 months ago
.gitignore	sync: jetbrains config	2 weeks ago
FyneApp.toml	fmt: logs and fyne app toml	2 months ago
LICENSE	Create LICENSE	10 months ago
Makefile	trying to patch docker on linux amd64	2 months ago
README.md	remove: call for participation from README	last month
crosscompile.sh	fix: ldflags	2 months ago
docker-compose.yml	trying to patch docker on linux amd64	2 months ago
flake.lock	Update Cargo lock and Rust deps	2 months ago
flake.nix	chore(rust-client): Update to latest master (#66)	last month

About

🥊 Components to measure Direct Connection Upgrade through Relay (DCUtR) performance.

- Readme
- Apache-2.0 license
- 32 stars
- 5 watching
- 9 forks

Releases 10

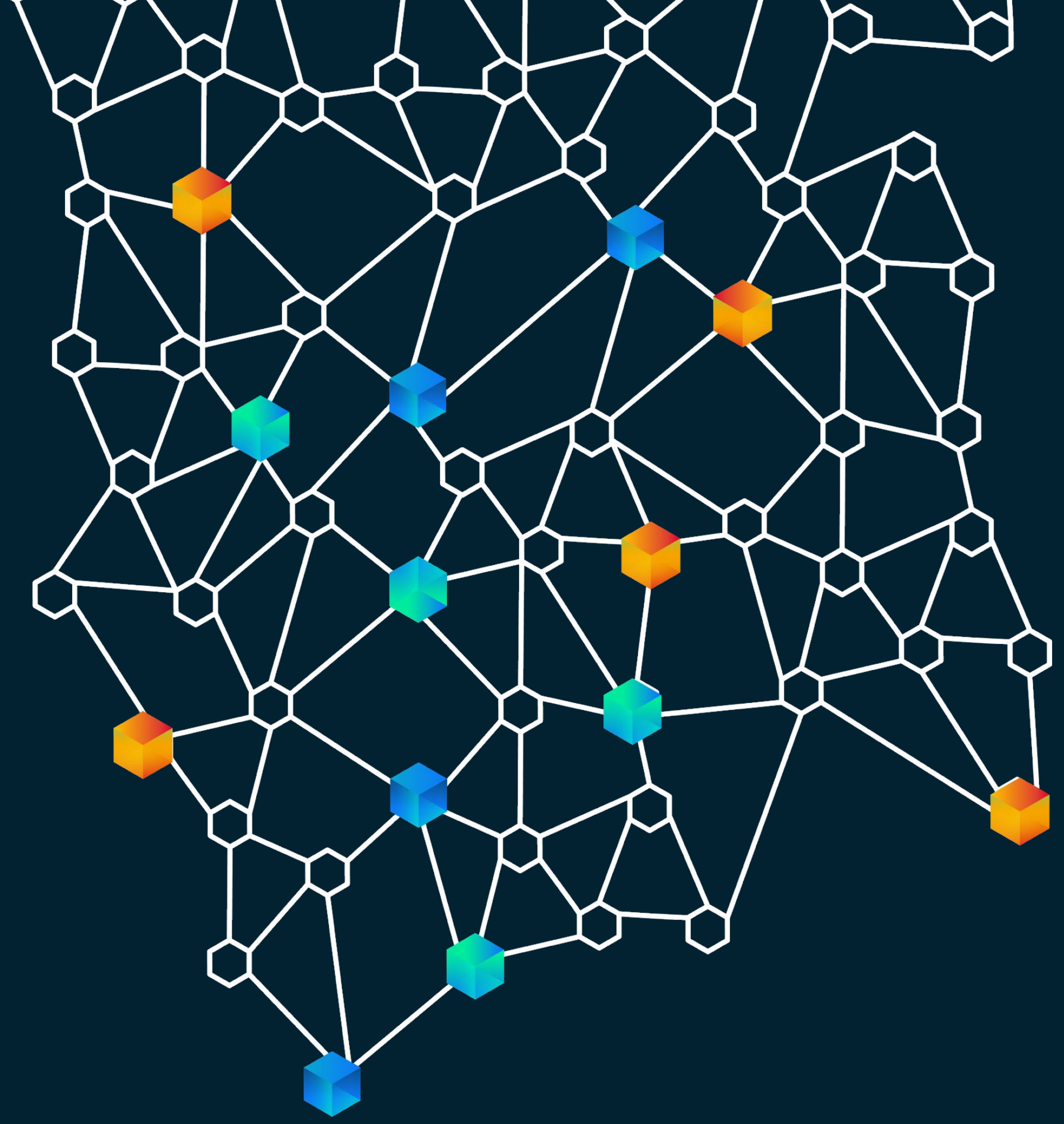
Hole Punch December Release Latest
on Nov 24, 2022

+ 9 releases

Contributors 7

Languages

● Jupyter Notebook 98.0%	● Go 1.4%
● Rust 0.3%	● PLpgSQL 0.1%
● Nix 0.1%	● Makefile 0.1%



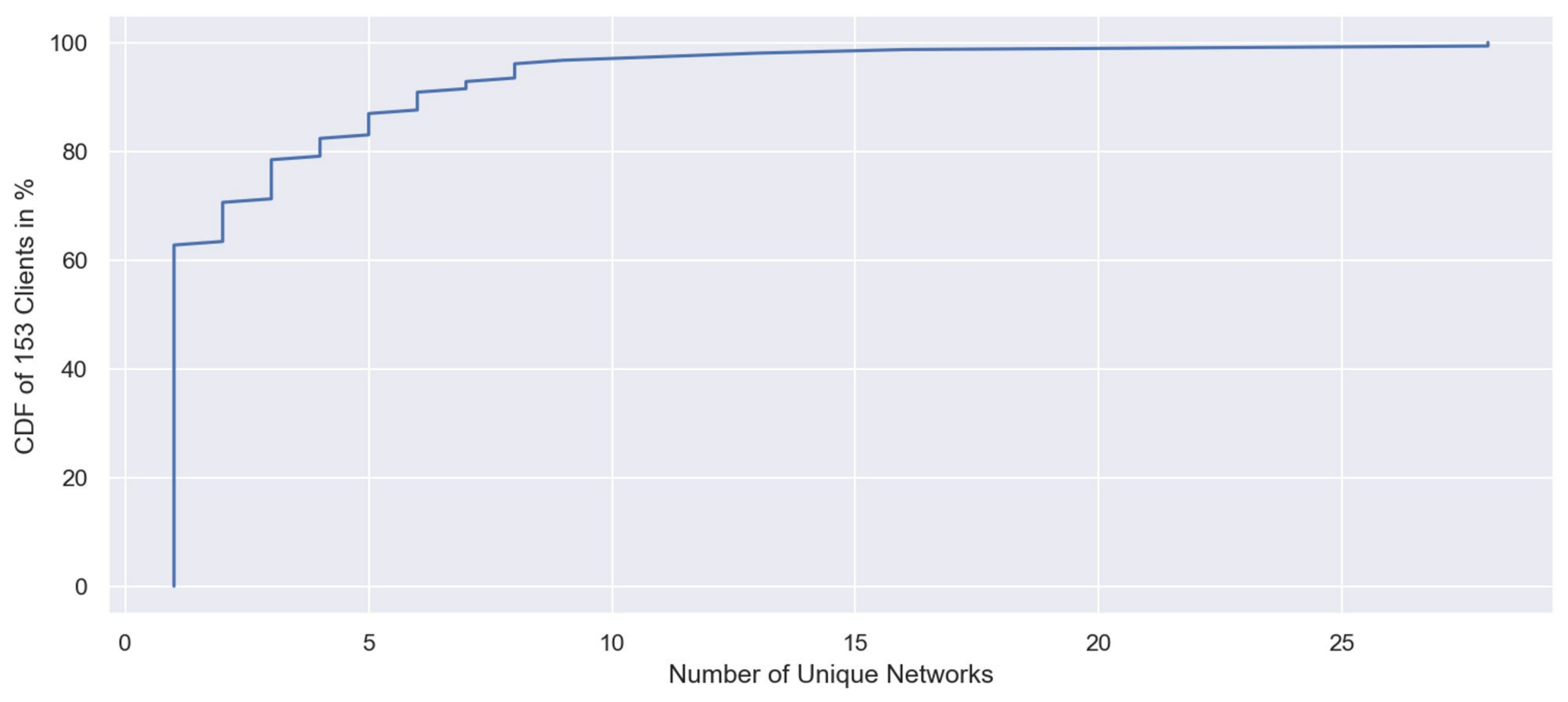
https://github.com/libp2p/punchr

Measurement Results

Network Detection Results

Statistics

- **153** Clients operated in **372** unique networks


CDF of 153 Clients in %
Number of Unique Networks