Hole Punching in IPFS/LibP2P

Dante Cullari

CEO - Konvergence Inc.

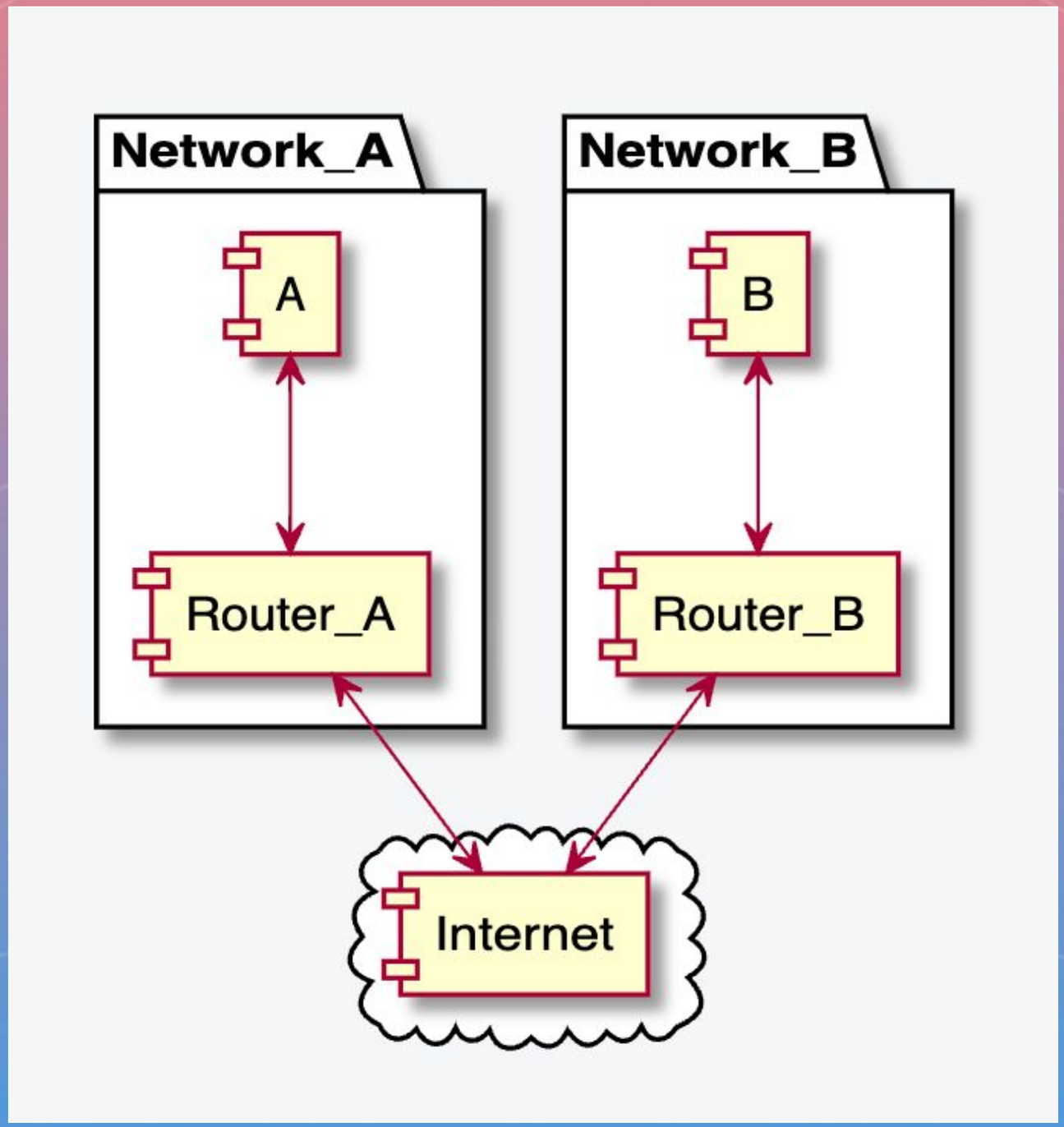
Web3 Architect, Full Stack Developer,
UI/UX Designer, Web Researcher

dante@raincloud.earth
https://cloudforest.cloud

Konvergence

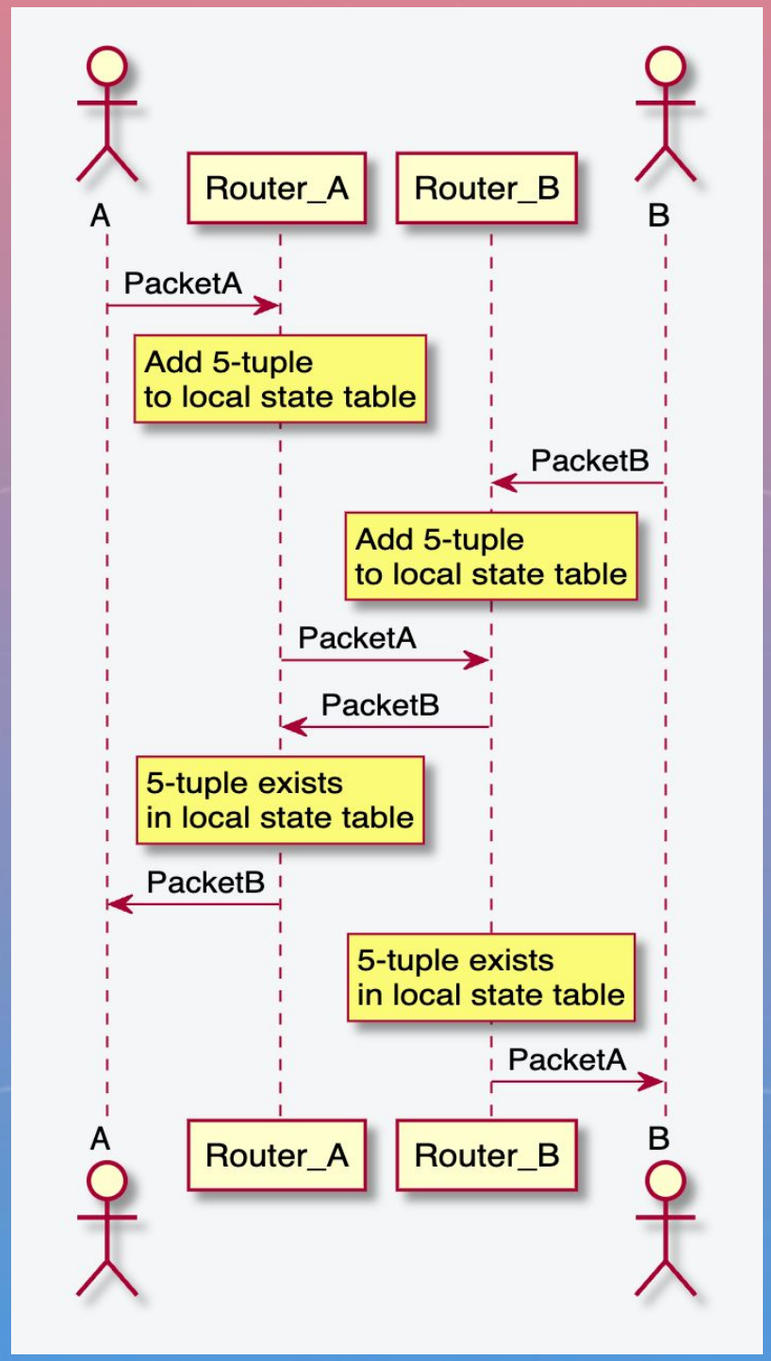
Cloud Forest

Standard Hole Punching



https://blog.ipfs.tech/2022-01-20-libp2p-hole-punching/

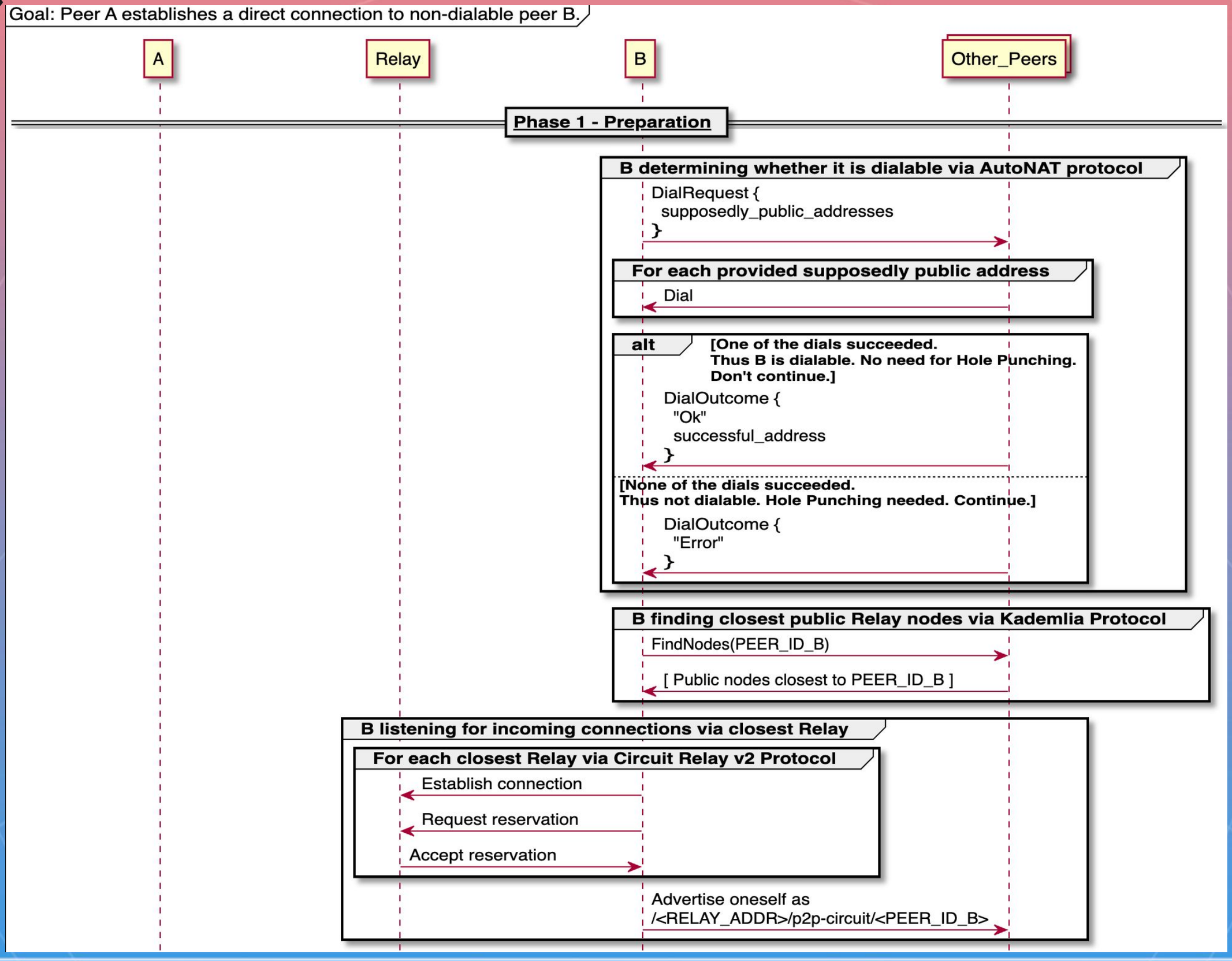
5-tuple (IP_A, IP_B, TCP, $Port_A$, $Port_B$)



https://blog.ipfs.tech/2022-01-20-libp2p-hole-punching/

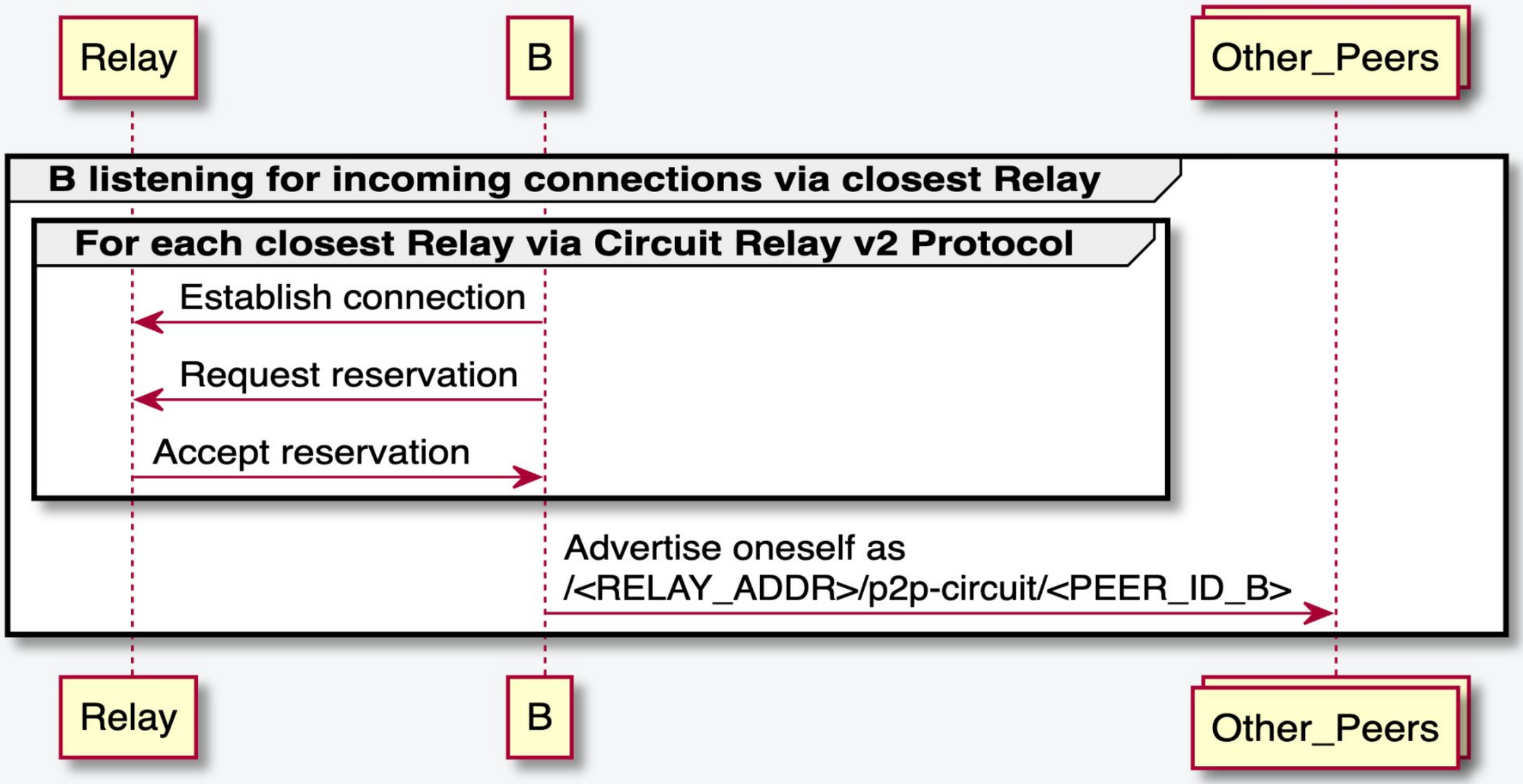
Decentralized Hole Punching

Preparation Phase - Using Relays



Relay Reservation

For each of the public relay nodes discovered in the previous step, ***B*** would do the following: First, ***B*** connects to the remote node and requests a so-called "reservation", basically saying: "Hey, I am not dialable. Given that you are dialable, would you mind listening for incoming connections on my behalf, forwarding each of them to me over this connection?".



Advertising Relayed Multi-Address

If the remote accepts the reservation request, ***B*** can advertise itself as being reachable through the remote Relay node. In other words, instead of advertising its own IP address (which is useless given that ***B*** is not publicly dialable), ***B*** advertises a "relayed" address, which contains the IP address of the remote relay node plus its own peer ID.

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

(The above is a so-called **multiaddr** . It is a composable network addressing schema. The address above reads as: "You can reach peer ***B*** with the peer ID `PEER_ID_B` via the relay at the address `RELAY_ADDR` ".)

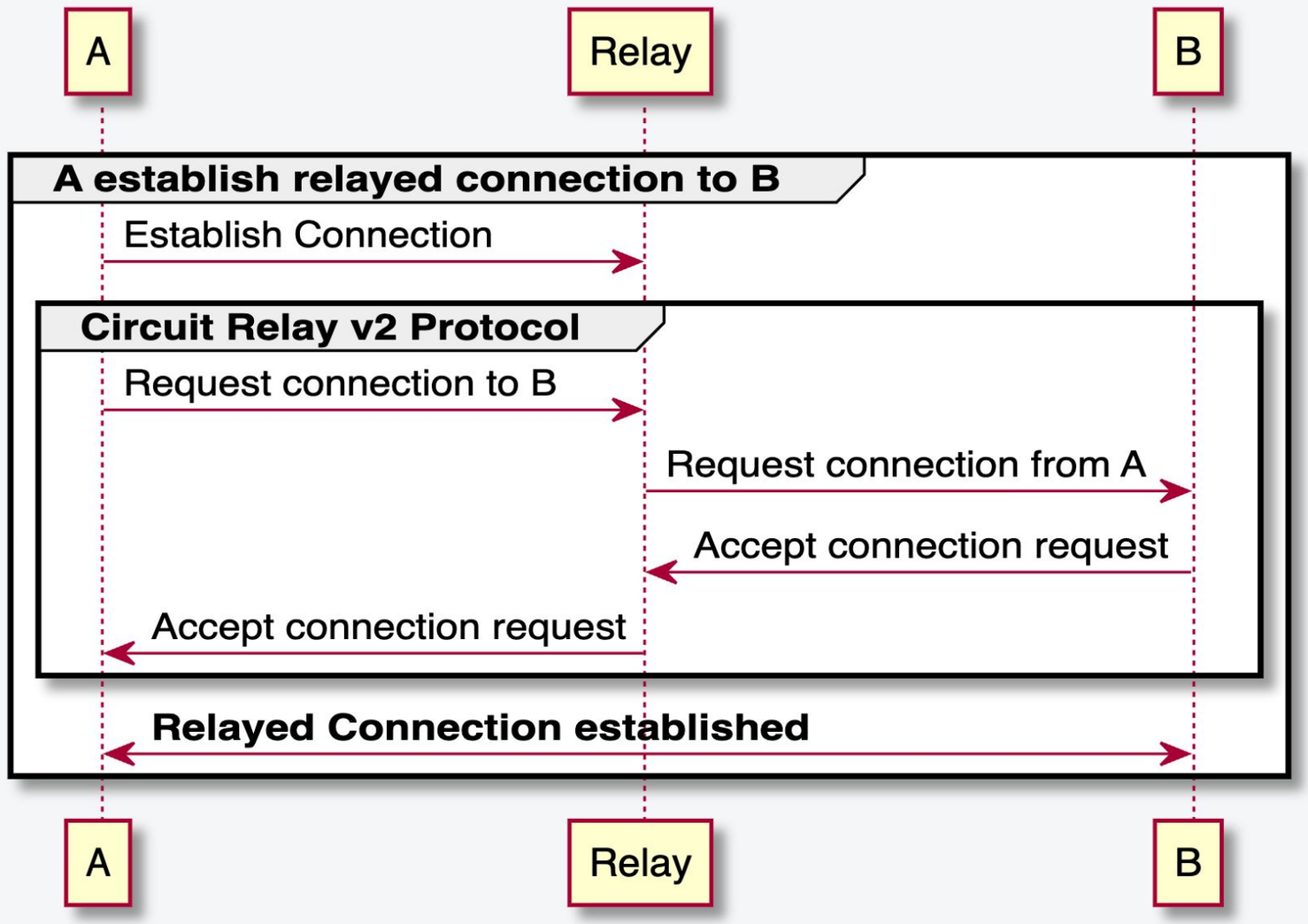
Note: It is very important that ***B*** keeps the outgoing connection to the relay node alive. ***B*** is not publicly dialable; thus, the relay would not be able to establish a connection to ***B***. When the relay receives a connection request for ***B***, it relays it using the initial connection from ***B***.

Secure Relayed Connection & Upgrading

2.1 Establish secured relayed connection (Circuit Relay v2)

Before establishing a direct connection using hole punching, ***A*** first has to establish a relayed connection to ***B*** via the public relay node. Using the information contained in ***B***'s advertised address, ***A*** first establishes a direct connection to the relay node and then requests a relayed connection to ***B*** from the relay. The relay forwards said request to ***B***, which accepts the request. The relay once more forwards the acceptance to ***A***. From now on, ***A*** and ***B*** can use the bi-directional channel over the relay to communicate.

A and ***B*** upgrade the relayed connection with a security protocol like TLS. Thus the relay cannot eavesdrop on the connection between the two.



Punching Holes

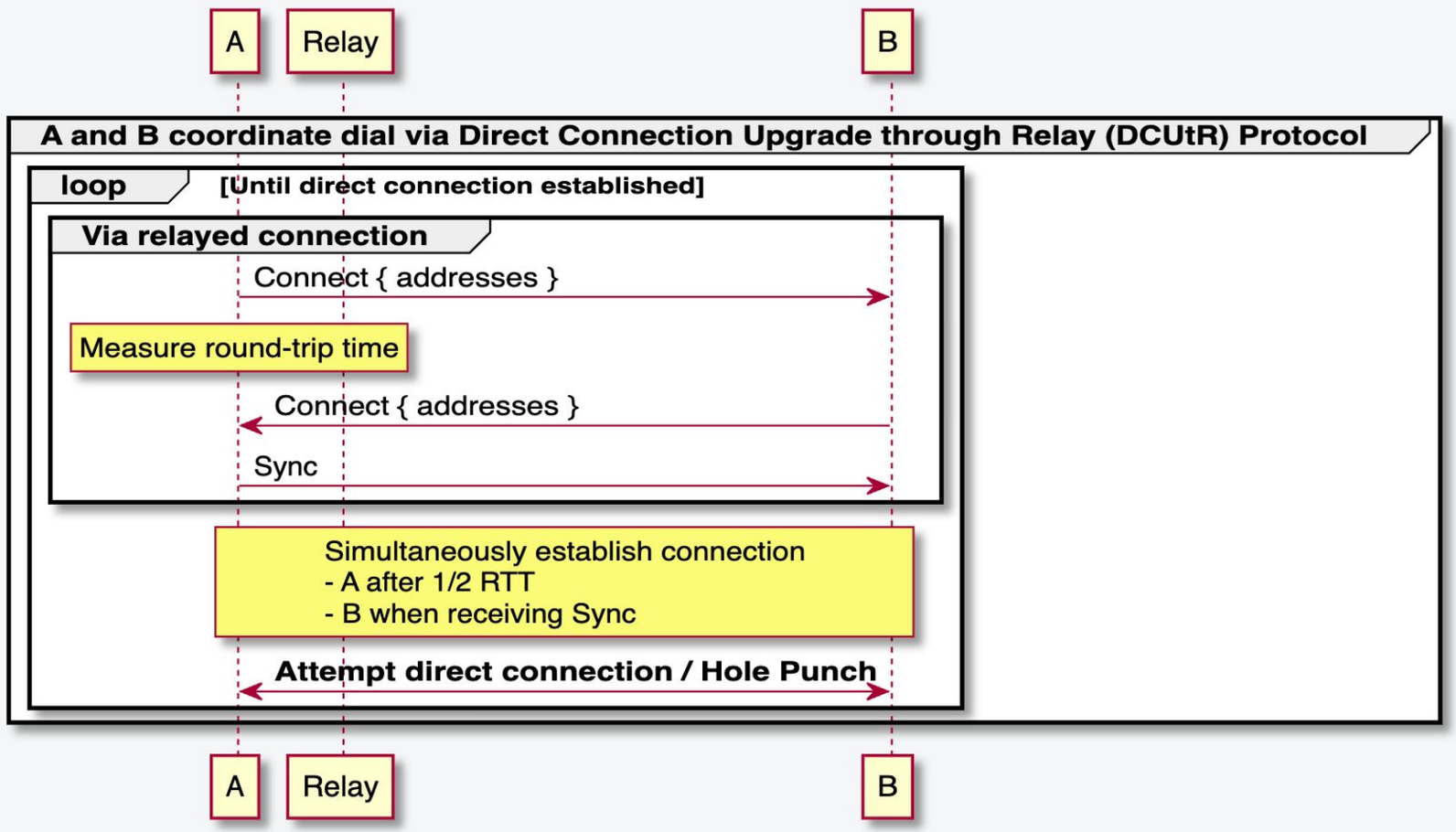
2.2. Coordinate simultaneous dial (DCUtR)

Over the relayed connection established in the previous step, ***A*** and ***B*** can now coordinate the hole punch, ultimately leading to a direct connection between ***A*** and ***B***. This coordination is happening via the libp2p DCUtR protocol which stands for "Direct Connection Upgrade through Relay" protocol. This is the magical synchronization mechanism, or rather a pretty good time synchronization mechanism.

There are two stages to do a ***direct connection upgrade through a relay***, exchanging ***Connect*** messages and sending a single ***Sync*** message.

First off, ***A*** sends a ***Connect*** message to ***B*** through the relay. That ***Connect*** message contains the addresses of ***A***. libp2p offers multiple mechanisms to discover one's addresses, e.g., via the libp2p identify protocol . ***B*** receives the ***Connect*** message on the relayed connection and replies with a ***Connect*** message containing its (non-relayed) addresses. ***A*** measures the time between sending its and receiving ***B***'s ***Connect*** message and thereby determines the round trip time between ***A*** and ***B*** (on the relayed connection).

Next, ***A*** sends a ***Sync*** message to ***B*** on the relayed connection. Once sent out, ***A*** waits for half the round trip time, then it dials ***B*** via the addresses received in ***B***'s ***Connect***. This is a direct dial not using the relayed connection. On the other end, as soon as ***B*** receives ***A***'s ***Sync*** message, it immediately directly dials ***A*** with the addresses provided in ***A***'s ***Connect*** message.



Other Implementations - Iroh

Orbit

https://iroh.computer

CAPITAL FACTORY

3rd Party Relay Network - Tailscale

Orbit

tailscale Pricing Use cases Enterprise Customers Download Blog Docs Log In Try for free

Search...

Docs › Reference › Technical overviews

DERP Servers

Tailscale runs DERP relay servers distributed around the world to link your Tailscale nodes peer-to-peer as a side channel during NAT traversal, and as a fallback in case NAT traversal fails and a direct connection cannot be established.

Because Tailscale private keys never leave the node where they were generated, there is never a way for a DERP server to decrypt your traffic. A DERP server just blindly forwards already-encrypted traffic from one node to another.

Tailscale runs DERP servers in many locations. As of September 2022, this list includes:

- Australia (Sydney)
- Brazil (São Paulo)
- Canada (Toronto)
- Dubai (Dubai)
- France (Paris)
- Germany (Frankfurt)
- Hong Kong (Hong Kong)
- India (Bangalore)

ON THIS PAGE

https://tailscale.com/kb/1232/derp-servers/

CAPITAL FACTORY

Open Source Multiplex Nodes - MASQUE Protocol

Orbit

The MASQUE Protocol

Abstract

This document describes MASQUE (Multiplexed Application Substrate over QUIC Encryption). MASQUE is a framework that allows concurrently running multiple networking applications inside an HTTP/3 connection. For example, MASQUE can allow a QUIC client to negotiate proxying capability with an HTTP/3 server, and subsequently make use of this functionality while concurrently processing HTTP/3 requests and responses.

This document is a straw-man proposal. It does not contain enough details to implement the protocol, and is currently intended to spark discussions on the approach it is taking. Discussion of this work is encouraged to happen on the MASQUE IETF mailing list masque@ietf.org or on the GitHub repository which contains the draft: https://github.com/DavidSchinazi/masque-drafts.

https://www.ietf.org/archive/id/draft-schinazi-masque-02.html

CAPITAL FACTORY

Orbit

Join us on Slack!

Continue Discussions online

Get Event updates

Reach out to Organizers

Orbit

Survey

- Filecoin Orbit Community Program
  Survey

Thank You

dante@raincloud.earth

https://raincloud.earth

https://twitter.com/KonvergenceInc