Introduction to and state of libp2p

A peer-to-peer networking library

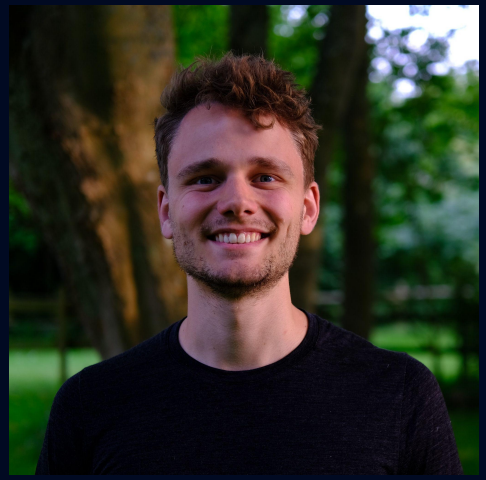
LIBP2P

Max Inden

Software Developer at Protocol Labs,
stewarding the libp2p project.

Maintainer of the Rust implementation.

mail@max-inden.de
@mxinden on GitHub / Twitter / …
https://max-inden.de



What is libp2p?

A modular peer-to-peer networking stack

- All you need to build peer-to-peer applications
- Composable building blocks based on a shared core to assemble future-proof p2p networking layers
- Implemented in 7+ languages
- Runs on many runtimes: browser, mobile, embedded
- Powers the IPFS, Ethereum 2, Filecoin and Polkadot network
- ~100_000 libp2p based nodes online at any given time

LIBP2P

Where?

Where does libp2p live?

L7 **Peer to Peer Application**

L3 / L4 **Transport**

L2 **Data-link Layer**

L1 **Physical Layer**

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

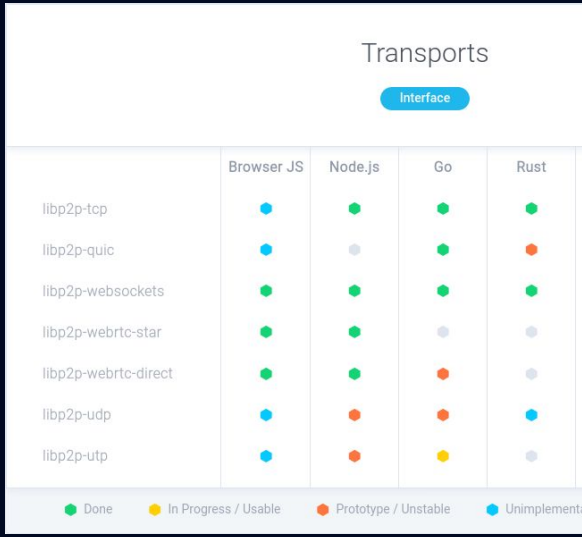
Messaging

Data Exchange

Transports

- Transports are core abstractions of libp2p
  - Enable connection establishment
  - Dialing and listening

- Current transports:
  - TCP
  - QUIC
  - WebSockets
- Experimental:
  - WebRTC
  - Bluetooth



Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

Secure Channels

- Peer authentication and transport encryption.
- Several security protocols supported:
  - Noise
  - TLS 1.3

noise-libp2p - Secure Channel Handshake

> A libp2p transport secure channel handshake built with the Noise Protocol Framework.

Lifecycle Stage	Maturity	Status	Latest Revision
3A	Recommendation	Active	r2, 2020-03-30

libp2p TLS Handshake

Lifecycle Stage	Maturity	Status	Latest Revision
2A	Candidate Recommendation	Active	r0, 2019-03-23

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

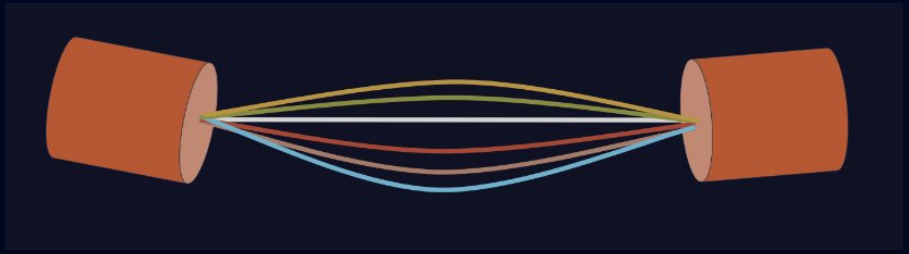
Routing

Messaging

Data Exchange

Multiplexing

- Establishing a P2P connection may not be cheap or easy (e.g. hole punching, negotiation, handshake, etc.)
- Re-use established connections for several protocols.
  - Applications can leverage already established connections.
- Several implementations of multiplexers available:
  - Language specific libraries for stream multiplex (Yamux, Mplex)
  - Transport protocol native multiplexing capabilities (QUIC)



Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

NAT Traversal

Motivation: *IPFS DHT crawl measurements (Nov 22nd 2019) showed that out of 4344 peers, 2754 were undialable (~63%).*

Goal:

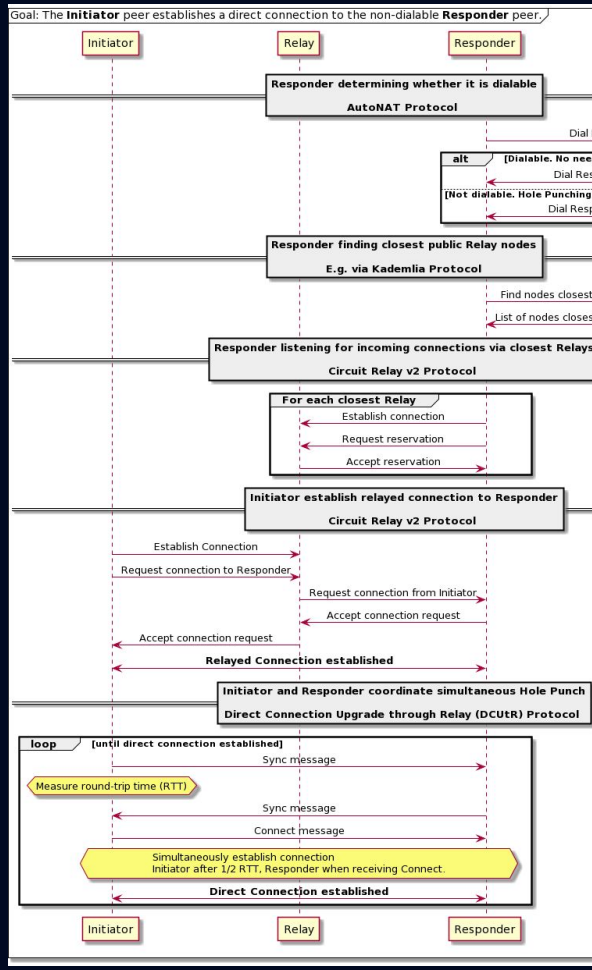
- Achieve global direct connectivity in heterogeneous networks.
- No dependency on central infrastructure.

NAT Traversal

Added in 2021

- Transport Protocols: TCP, QUIC
- Relay Protocol (TURN-like): Circuit Relay v2
- Signaling Protocol: Direct Connection Upgrade through Relay (DCUtR)
- STUN-like Protocol: AutoNAT

Next up: use this in WebRTC



Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

Peer Discovery

- Discover random peers (supporting certain services)
- Implementations
  - mDNS (Multicast DNS)
  - Rendezvous
  - GossipSub peer exchange

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

Routing - Kademlia DHT

- Distributed hash table
- Based on the Kademlia paper
- Operations:
  - FIND_NODE
  - GET_VALUE and PUT_VALUE
  - GET_PROVIDER and PUT_PROVIDER

Kademlia: A Peer-to-peer Information System Based on the XOR Metric

Petar Maymounkov and David Mazières
{petar,dm}@cs.nyu.edu
http://kademlia.scs.cs.nyu.edu

New York University

Abstract. We describe a peer-to-peer distributed hash table with provable consistency and performance in a fault-prone environment. Our system routes queries and locates nodes using a novel XOR-based metric topology that simplifies the algorithm and facilitates our proof. The topology has the property that every message exchanged conveys or reinforces useful contact information. The system exploits this information to send parallel, asynchronous query messages that tolerate node failures without imposing timeout delays on users.

https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

Messaging - GossipSub

- Publish and subscribe
- Brokerless, self-regulating, no global knowledge
- Eager push and lazy pull

GossipSub: Attack-Resilient Message Propagation the Filecoin and ETH2.0 Networks

Dimitris Vyzovitis
Protocol Labs
vyzo@protocol.ai

Yusef Napora
Protocol Labs
yusef@protocol.ai

Dirk McCormick
Protocol Labs
dirk@protocol.ai

David Dias
Protocol Labs
david@protocol.ai

Yiannis Psaras
Protocol Labs
yiannis@protocol.ai

ABSTRACT

Permissionless blockchain environments necessitate the use of a fast and attack-resilient message propagation protocol for Block and Transaction messages to keep nodes synchronised and avoid forks. We present GossipSub, a gossip-based pubsub protocol, which, in contrast to past pubsub protocols, incorporates resilience against a wide spectrum of attacks

ACM Reference Format:
Dimitris Vyzovitis, Yusef Napora, Dirk McCormick, Da
and Yiannis Psaras. 2020. GossipSub: Attack-Resilient Mess
agation in the Filecoin and ETH2.0 Networks. In *Proceedin*
tocol Labs TechRep (PL-TechRep-2020-002). Protocol Labs,
https://doi.org/10.1145/nnnnnnn.nnnnnnn

https://arxiv.org/pdf/2007.02754.pdf

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

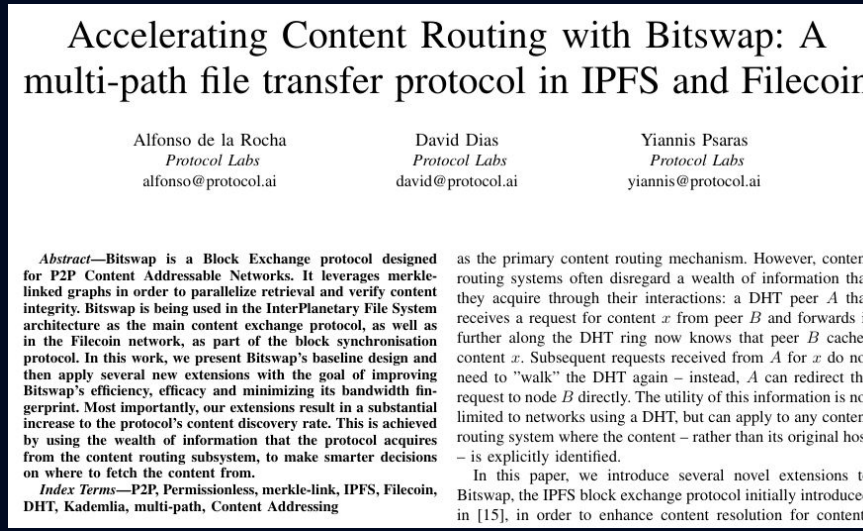
Routing

Messaging

Data Exchange

Data Exchange - Bitswap

- Message-oriented protocol
- Exchange blocks of data
  - Requests
    - WANT-HAVE
    - WANT-BLOCK
    - CANCEL
  - Responses
    - HAVE
    - BLOCK
    - DONT_HAVE



Accelerating Content Routing with Bitswap: A multi-path file transfer protocol in IPFS and Filecoin

Alfonso de la Rocha
Protocol Labs
alfonso@protocol.ai

David Dias
Protocol Labs
david@protocol.ai

Yiannis Psaras
Protocol Labs
yiannis@protocol.ai

***Abstract*—Bitswap is a Block Exchange protocol designed for P2P Content Addressable Networks. It leverages merkle-linked graphs in order to parallelize retrieval and verify content integrity. Bitswap is being used in the InterPlanetary File System architecture as the main content exchange protocol, as well as in the Filecoin network, as part of the block synchronisation protocol. In this work, we present Bitswap's baseline design and then apply several new extensions with the goal of improving Bitswap's efficiency, efficacy and minimizing its bandwidth fingerprint. Most importantly, our extensions result in a substantial increase to the protocol's content discovery rate. This is achieved by using the wealth of information that the protocol acquires from the content routing subsystem, to make smarter decisions on where to fetch the content from.**

***Index Terms*—P2P, Permissionless, merkle-link, IPFS, Filecoin, DHT, Kademlia, multi-path, Content Addressing**

as the primary content routing mechanism. However, conter
routing systems often disregard a wealth of information tha
they acquire through their interactions: a DHT peer A tha
receives a request for content x from peer B and forwards
further along the DHT ring now knows that peer B cache
content x. Subsequent requests received from A for x do no
need to "walk" the DHT again – instead, A can redirect th
request to node B directly. The utility of this information is no
limited to networks using a DHT, but can apply to any conter
routing system where the content – rather than its original hos
– is explicitly identified.

In this paper, we introduce several novel extensions t
Bitswap, the IPFS block exchange protocol initially introduce
in [15], in order to enhance content resolution for content

https://arxiv.org/pdf/2007.02754.pdf

Transports

Secure Channels

Multiplexers

NAT Traversal

LIBP2P

Discovery

Routing

Messaging

Data Exchange

libp2p Implementations

go-libp2p Public

libp2p implementation in Go

Go 4,166 MIT 651 170 (8 issues need help) 13 Updated 30 minutes ago

rust-libp2p Public

The Rust Implementation of the libp2p networking stack.

Rust 2,127 MIT 426 94 (9 issues need help) 21 Updated 18 hours ago

js-libp2p Public

The JavaScript Implementation of libp2p networking stack.

JavaScript 1,705 MIT 315 145 (18 issues need help) 33 Updated 1 hour ago

cpp-libp2p Public

C++17 implementation of libp2p

C++ 177 44 15 2 Updated yesterday

jvm-libp2p Public

a libp2p implementation for the JVM, written in Kotlin 🔥 [WIP]

Kotlin 131 53 20 (2 issues need help) 1 Updated 4 days ago

nim-libp2p Public

libp2p implementation in Nim

Nim 146 29 39 (2 issues need help) 22 Updated 19 hours ago

py-libp2p Public

The Python implementation of the libp2p networking stack 🐍 [under development]

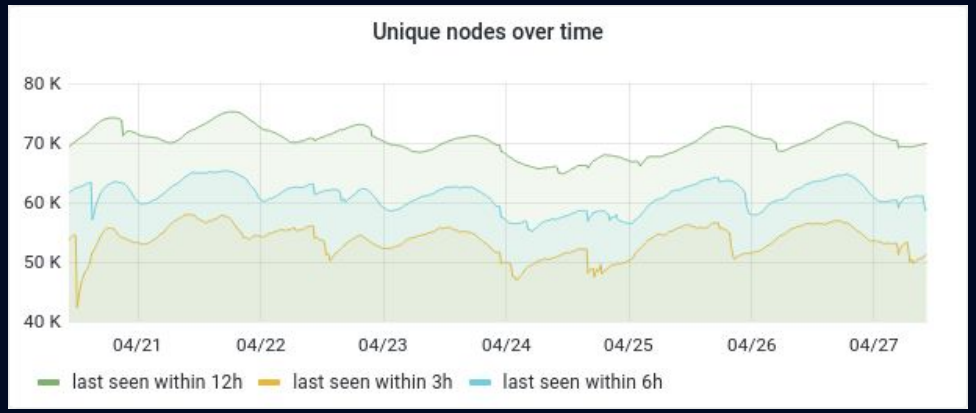
Python 311 73 55 (2 issues need help) 6 Updated on Mar 17, 2021

erlang-libp2p Public

An Erlang implementation of libp2p swarms

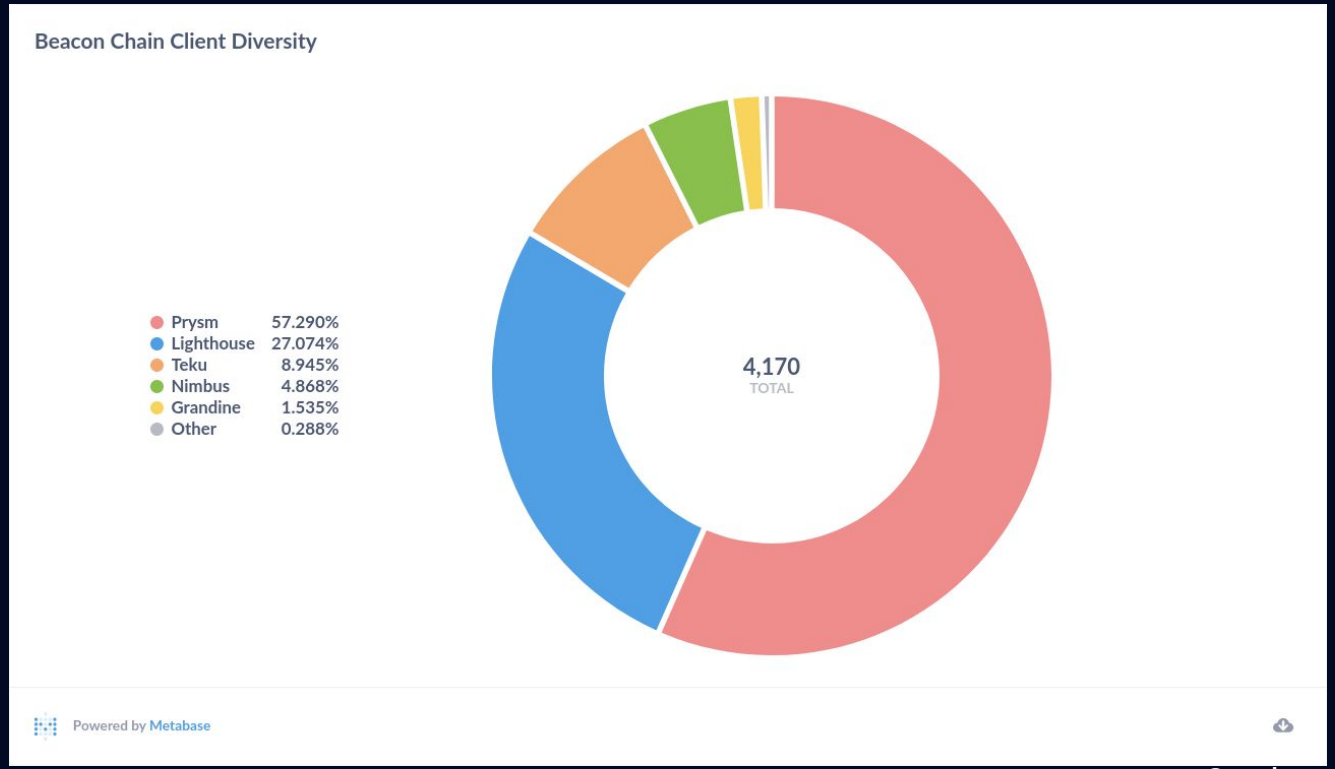
Projects using libp2p

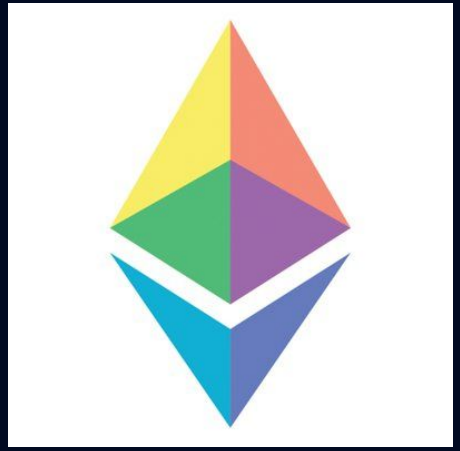
IPFS



Explore the data via
https://kademlia-exporter.max-inden.de/

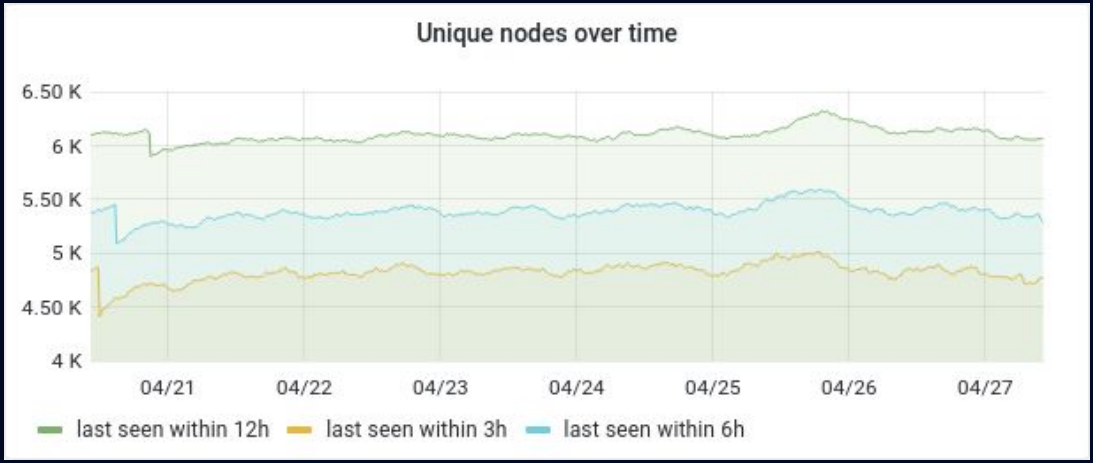
Ethereum 2

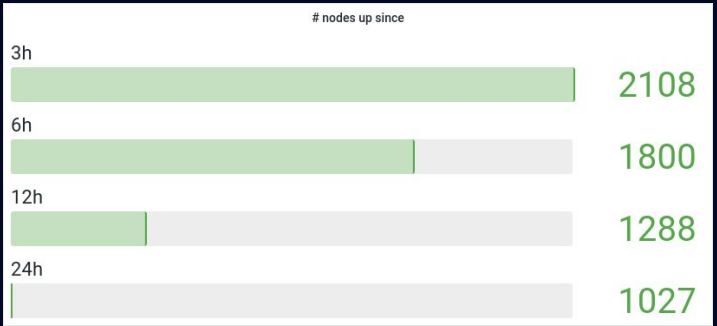




See https://migalabs.es/beaconnodes

Filecoin





Explore the data via
https://kademlia-exporter.max-inden.de/

Polkadot

Explore the data via
https://kademlia-exporter.max-inden.de/

Berty

- Offline-first
- Peer-to-peer
- Messaging app

Where is libp2p heading?

Roadmap

- 🕸️ Unprecedented global connectivity
- 🤝 Low latency, efficient connection handshake via Protocol Select
- 🌐 Improved browser connectivity

More details:
https://github.com/libp2p/specs/blob/master/ROADMAP.md


Transports
Secure Channels
Multiplexers
Peer Discovery
LIBP2P
Peer Routing
Content Routing
NAT Traversal
Pubsub

How to 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

mail@max-inden.de
@mxinden on GitHub / Twitter / ...
https://max-inden.de