State of libp2p

Status quo and future roadmap of the 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
Where does libp2p live?

- L7 Peer to Peer Application
- L3 / L4 Transport
- L2 Data-link Layer
- L1 Physical Layer
LIBP2P

- Transports
- Secure Channels
- Multiplexers
- NAT Traversal
- Discovery
- Routing
- Messaging
- Data Exchange
LIBP2P
Transports

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

- Current transports:
  - TCP
  - QUIC
  - WebSockets

- Experimental:
  - WebRTC
  - Bluetooth
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.

<table>
<thead>
<tr>
<th>Lifecycle Stage</th>
<th>Maturity</th>
<th>Status</th>
<th>Latest Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3A</td>
<td>Recommendation</td>
<td>Active</td>
<td>r2, 2020-03-30</td>
</tr>
</tbody>
</table>

libp2p TLS Handshake

<table>
<thead>
<tr>
<th>Lifecycle Stage</th>
<th>Maturity</th>
<th>Status</th>
<th>Latest Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>2A</td>
<td>Candidate Recommendation</td>
<td>Active</td>
<td>r0, 2019-03-23</td>
</tr>
</tbody>
</table>
LIBP2P

- Transports
- Secure Channels
- Multiplexers
- NAT Traversal
- 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)
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 - Project Flare
- 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
Peer Discovery

- Discover random peers (supporting certain services)
- Implementations
  - mDNS (Multicast DNS)
  - Rendezvous
  - GossipSub peer exchange
Transports
Secure Channels
Multiplexers
NAT Traversal
Discovery
Routing
Messaging
Data Exchange
LIBP2P
Routing - Kademlia DHT

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

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

Petar Maymounkov and David Mazieres
{petar, dm}@cs.nyu.edu
http://kademiia.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 tightness delays on users.
Messaging - GossipSub

- Brokerless, self-regulating, no global knowledge
- Publish and subscribe
- Eager push and lazy pull
- Use cases: IPNS, content-addressing, blockchain consensus, message dissemination, etc.

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

Dimitris Vyzovitis  
Protocol Labs  
vzyo@protocol.ai

Yusef N Rapora  
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 attack scenarios.

ACM Reference Format:

LIBP2P

Transports
Secure Channels
Multiplexers
NAT Traversal
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 Pitas
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

libp2p
Implementations
**go-libp2p**  
Public  
libp2p implementation in Go

**rust-libp2p**  
Public  
The Rust Implementation of the libp2p networking stack.

**js-libp2p**  
Public  
The JavaScript Implementation of libp2p networking stack.
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/

Unique nodes over time

- last seen within 12h
- last seen within 3h
- last seen within 6h

<table>
<thead>
<tr>
<th># nodes up since</th>
<th>18853</th>
</tr>
</thead>
<tbody>
<tr>
<td>3h</td>
<td>15066</td>
</tr>
<tr>
<td>6h</td>
<td>10579</td>
</tr>
<tr>
<td>12h</td>
<td>6260</td>
</tr>
<tr>
<td>24h</td>
<td></td>
</tr>
</tbody>
</table>
Ethereum 2

See https://www.nodewatch.io/
Filecoin

Explore the data via [https://kademlia-exporter.max-inden.de/](https://kademlia-exporter.max-inden.de/)
Polkadot

Explore the data via https://kademia-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
- 🌐 Browser connectivity via WebRTC

More details:
https://github.com/libp2p/specs/blob/master/ROADMAP.md
LIBP2P

- Transports
- Secure Channels
- Multiplexers
- Peer Discovery
- NAT Traversal
- Peer Routing
- Content Routing
- Pubsub
Thank you for joining

- Documentation - docs.libp2p.io/
- Forum - discuss.libp2p.io/
- Specification & Roadmap - github.com/libp2p/specs/