libp2pによるP2P通信



https://libp2p.io/

https://docs.libp2p.io/

https://docs.rs/libp2p/latest/libp2p/

https://github.com/libp2p/rust-libp2p

spec 仕様のようなもの？

https://deepwiki.com/libp2p/rust-libp2p

P2P 通信のフレームワークでいろいろな言語向けにリリースされている。

元は IPFSで開発されていた。

いろいろできるのだけど、どこから読み始めたらよいのかわからない。。。

tracingによるログ出力

tracing_subscriber を実装しておいて、実行環境で RUST_LOG を設定してから実行するとトレースログが出力される。

libp2p が内部でやってくれているのでアプリ側でログ出力しなくても勝手に出力される。

```
tracing
   use tracing_subscriber::EnvFilter;

   fn main() ... {
       let _ = tracing_subscriber::fmt()
           .with_env_filter(EnvFilter::from_default_env())
```

```
        .try_init();
```

ログの例

```
$ ./target/debug/chat
2025-11-15T10:01:49.062909Z  INFO libp2p_swarm:
local_peer_id=12D3KooWSLEvyDedqxjR97QmvLeTRkveCVv9pDr1DvyxCgXcTcQU
Enter messages via STDIN and they will be sent to connected peers using Gossipsub
2025-11-15T10:01:49.121562Z TRACE NetworkBehaviour::poll:
netlink_proto::connection: polling Connection
2025-11-15T10:01:49.121613Z TRACE NetworkBehaviour::poll:
netlink_proto::connection: reading incoming messages
2025-11-15T10:01:49.121661Z TRACE NetworkBehaviour::poll:
netlink_proto::connection: poll_read_messages called
2025-11-15T10:01:49.121707Z TRACE NetworkBehaviour::poll:
netlink_proto::connection: polling socket
......
```

ping

Ping Tutorial

deepwiki

2つ以上プロセスを立ち上げて使う。
プロセスを立ち上げると標準出力に待ち受けているIPアドレスとポート番号を出力する。
もう1つのプロセスで接続先を指定するとそこに ping を投げる(15秒間隔)。
受信タイムアウトするまで ping-pong を繰り返す。

- swarm というのが基盤になる
  - Transport(TCP, UDPなどの接続方法や、TLS, noiseなどの暗号化レイヤー、Yamux,Mplexなどのストリーム多重化)
    - セキュリティやストリーム多重化は必須
      - 実験で暗号化したくないときのために plaintext transport はある
  - Behaviour(どういう振る舞いをするか)
    - このサンプルでは `ping` を使うが、これは ICMP の ping ではなく IPFSのping
    - 15秒ごとに送信、受信タイムアウトは20秒(Config)
    - 中身が実装されているのでここでは指定するだけ
  - swam の設定変更
    - デフォルトでは受信タイムアウトが10秒になっているので、接続直後の ping-pong して10秒後に切断される

切断後に再接続すれば

自身の `peerId` は keypair の公開鍵からハッシュ計算などして得られる文字列

`swarm.select_next_some()` で何かイベントがあるまで待つ

`loop{〜}` でぐるぐる

chat

https://github.com/libp2p/rust-libp2p/tree/master/examples/chat

mDNS

gossipsub protocol