Composable DHT

Enabling More Applications to Join the libp2p DHT Ecosystem

Gui Michel
@guissou

Probelab,
Protocol Labs

IPFS ÞING

Protocol Labs

IPFS Thing
15th April 2023

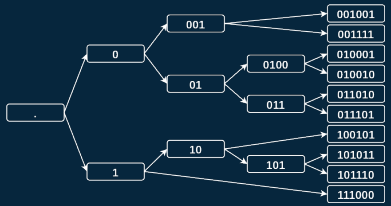
Disclaimer

- Very early design prototype
- Gather communuity feedback

Here be dragons

Kademlia DHT

- A Distributed Hash Table (DHT) is a decentralized overlay network used for peer and content routing
- Each node is uniquely identified by a binary `Key`, determining the node's position in the Keyspace and the distance between nodes
- Routing table groups nodes into *buckets* based on their distance from the local node's `Key`, bucket `i` containing peers `ID` sharing a `i` bit prefix
- Iterative lookups to efficiently find peers and content in the network, making it highly scalable



Kademlia Binary Trie

Libp2p Kademlia DHT Protocol

- IPFS-centric Kademlia implementations
- Kad DHT is the peer routing component of Libp2p
- Not future proof

- The implementations are correct but the protocol is flawed

Building a P2P messaging app using Libp2p

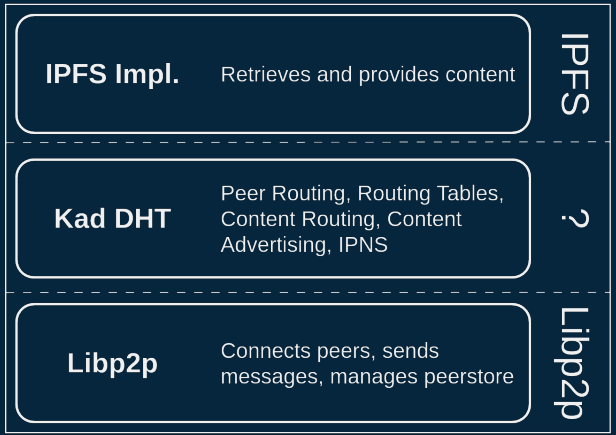
- Routing and Hole Punching using the DHT as peer discovery mechanism
- Feature: Rendez-vous for group chats
- Don’t want nodes to store and serve IPFS data
- If participating in the DHT but not serving IPFS data, IPFS will be hurt

- Other apps cannot really use the Libp2p DHT, they have to implement their own

Current DHT stack

IPFS Impl.	Retrieves and provides content
Kad DHT	Peer Routing, Routing Tables, Content Routing, Content Advertising, IPNS
Libp2p	Connects peers, sends messages, manages peerstore

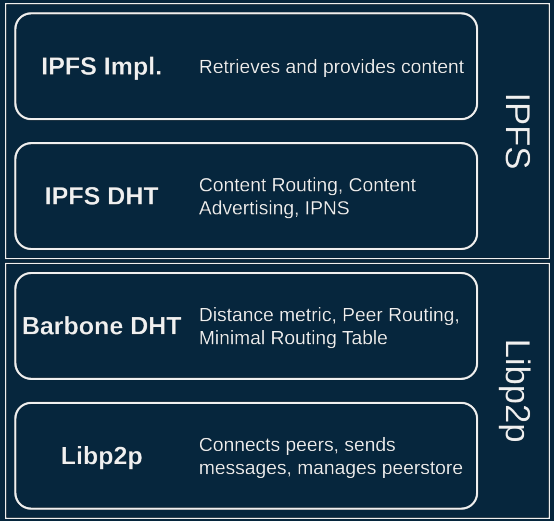
Current DHT stack



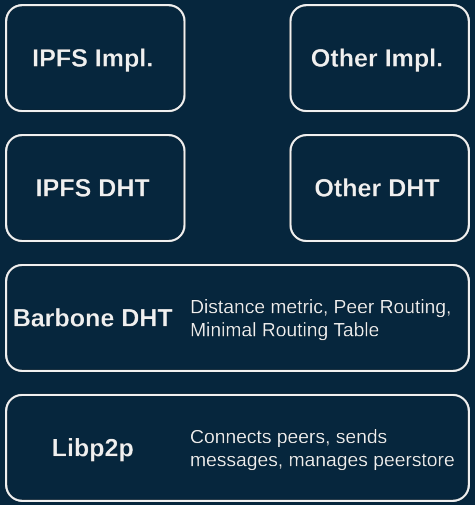
Improved DHT stack

IPFS Impl.	Retrieves and provides content
IPFS DHT	Content Routing, Content Advertising, IPNS
Barbone DHT	Distance metric, Peer Routing, Minimal Routing Table
Libp2p	Connects peers, sends messages, manages peerstore

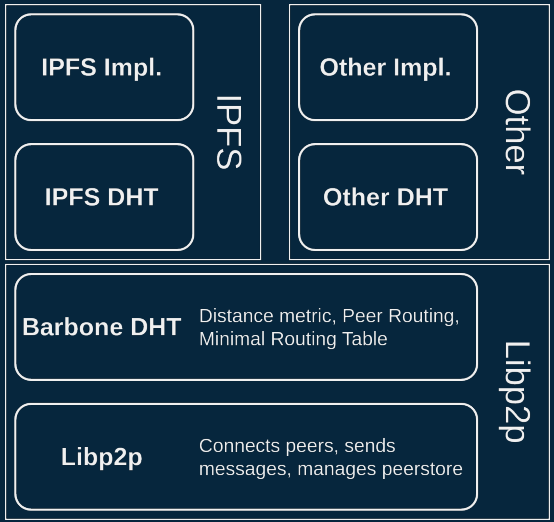
Improved DHT stack



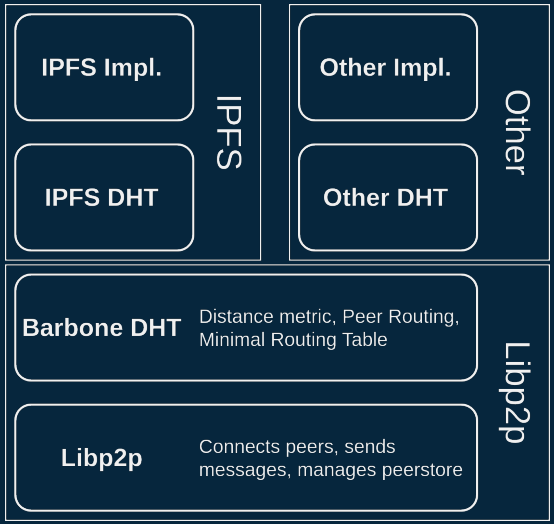
Improved DHT stack



Improved DHT stack



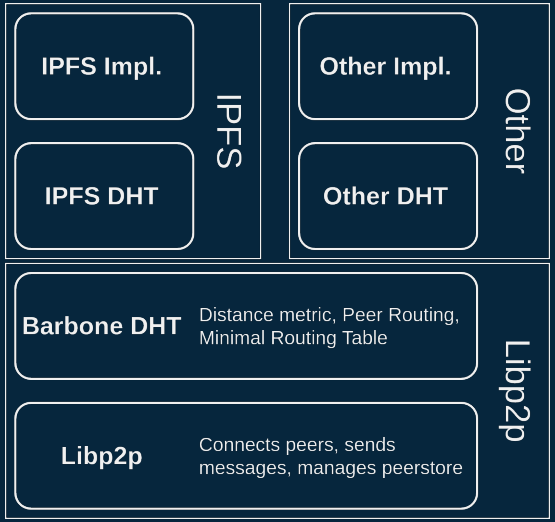
But wait, it doesn't work!



- IPFS `Provide` won't exactly work
- IPFS `FindProvs` will not be accurate

- Unless we change the `IPFS DHT` Protocol!

Adapting the IPFS DHT Protocol



- IPFS content should be allocated to IPFS nodes only
- IPFS content should be looked up on IPFS nodes only
- Same for Other DHT's RPCs
- Peer Routing still works like before

- How do we segregate applications?

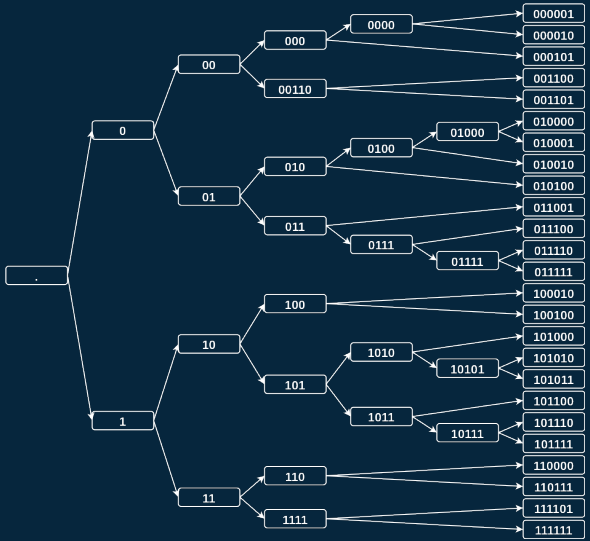
Features & Protocols

- A *Feature* is defined as support for a specific RPC
- A *Protocol* is defined as a set of features
- In a Protocol, Features must be ordered according to their importance
- Each node advertises its ordered set of features when communicating with a remote peer
- The features are recorded in the Routing Table for each remote peer

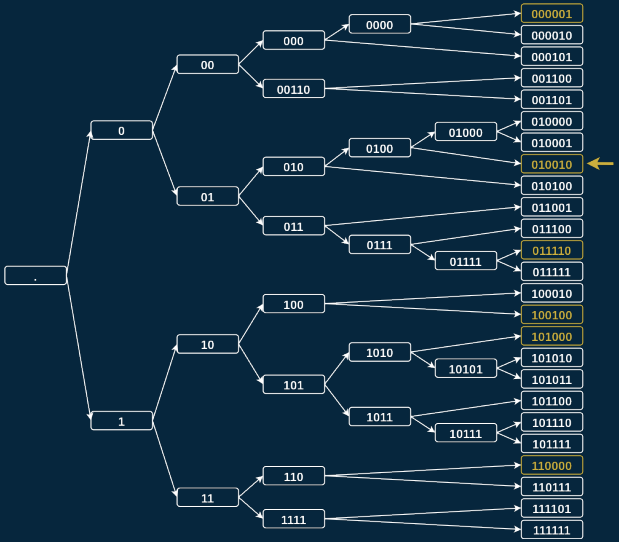
Routing Table Peer Selection Process

- Currently based on seniority only
- New selection criteria
  1. Features preferences
  2. Seniority

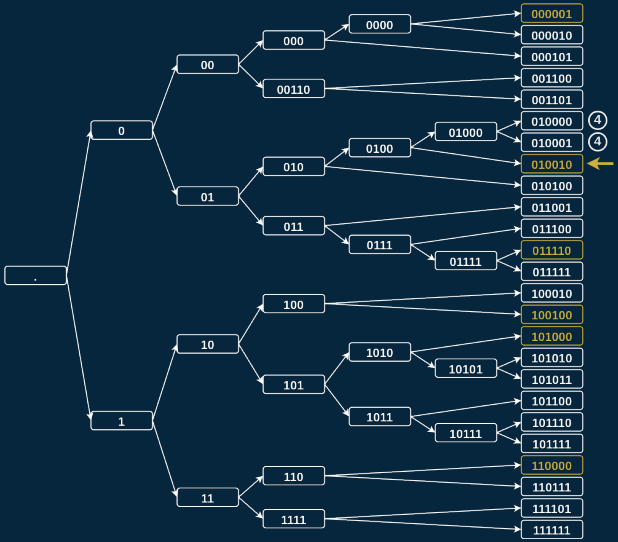
Routing Table Peer Selection Example



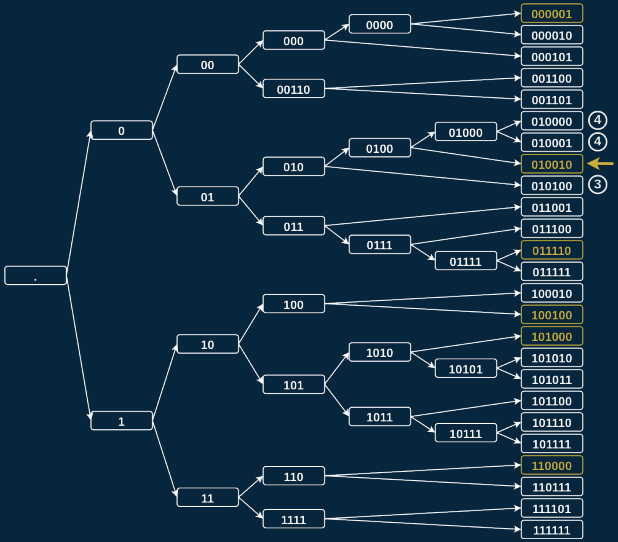
Routing Table Peer Selection Example



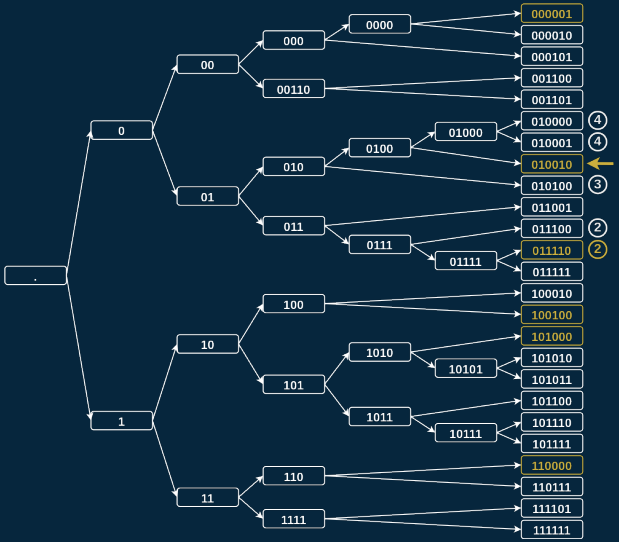
Routing Table Peer Selection Example



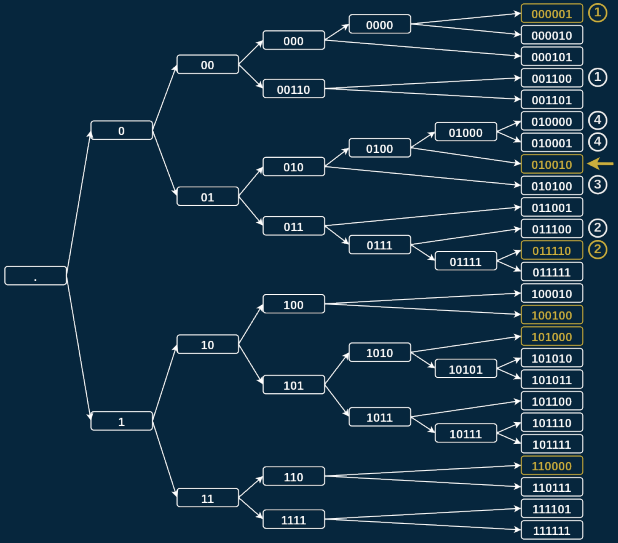
Routing Table Peer Selection Example



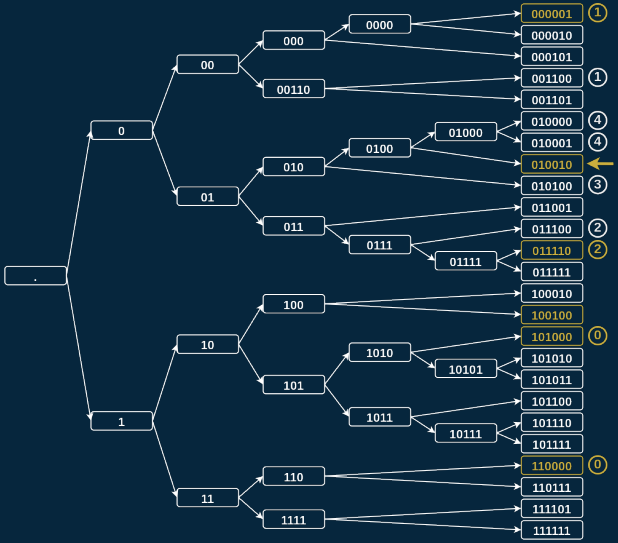
Routing Table Peer Selection Example



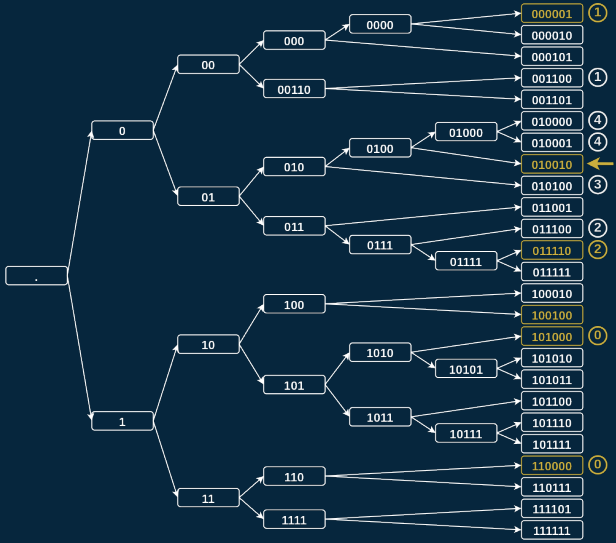
Routing Table Peer Selection Example



Routing Table Peer Selection Example



Routing Table Peer Selection Example



Routing Table

Bucket 0
101000
110000

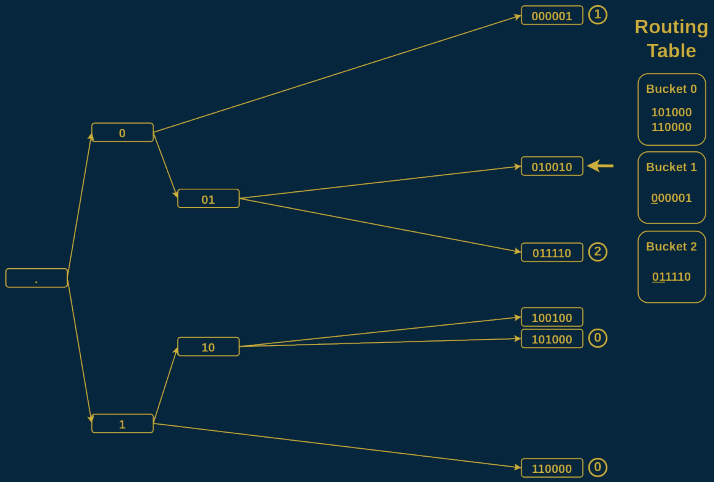
Bucket 1
000001
001100

Bucket 2
011100
011110

Bucket 3
010100

Bucket 4
010000
010001

Routing Table Peer Selection Example



Query process

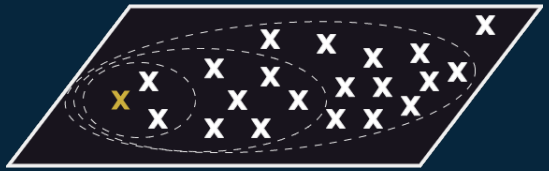
- In peer lookup, indicate your features priorities
- Requested peers return in priority peers matching the features priorities
- Requested peers' routing table is expected to have the same features

- Feature specific RPC only stays in the feature's sub-DHT

Routing Table Multi-Features Peers Selection

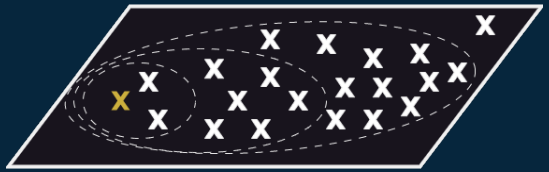
Kademlia keyspace 2D distance projection

Routing Table Multi-Features Peers Selection

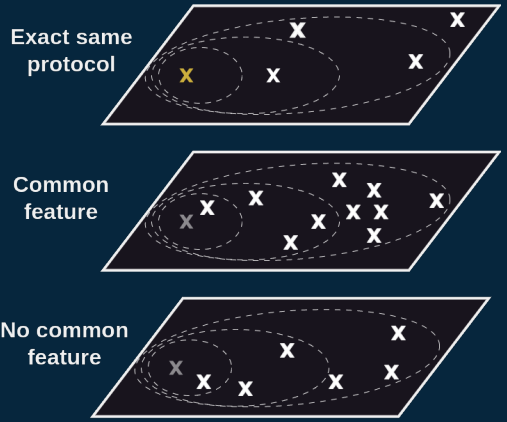


Kademlia keyspace 2D distance projection

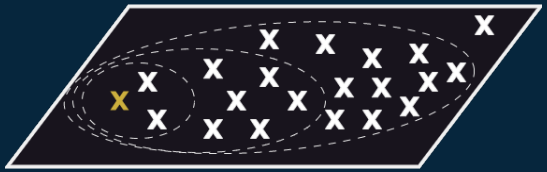
Routing Table Multi-Features Peers Selection



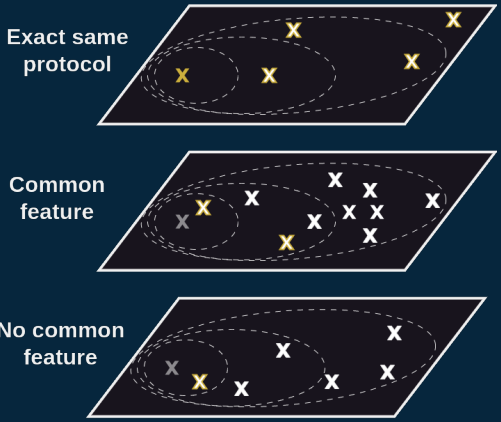
Kademlia keyspace 2D distance projection



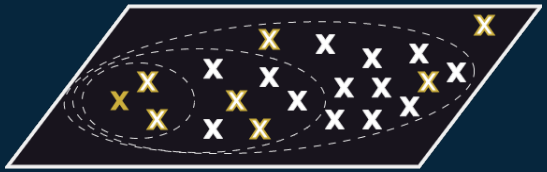
Routing Table Multi-Features Peers Selection



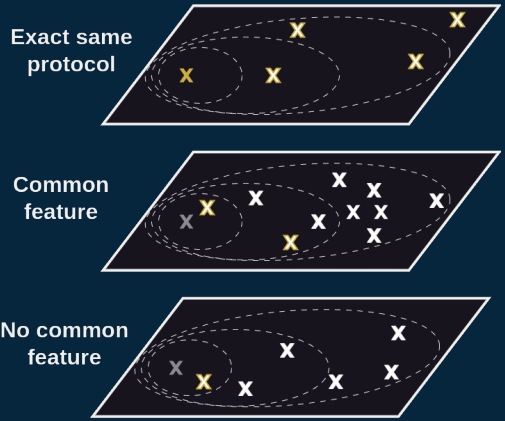
Kademlia keyspace 2D distance projection



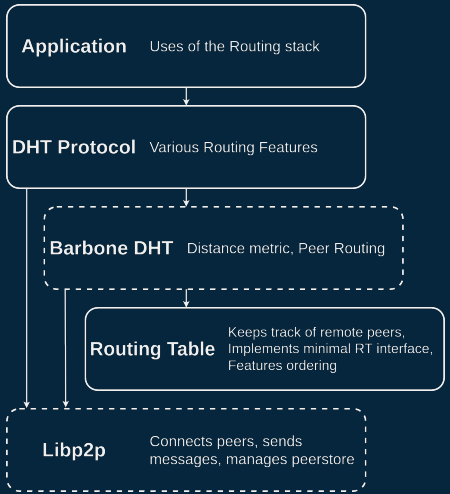
Routing Table Multi-Features Peers Selection



Kademlia keyspace 2D distance projection



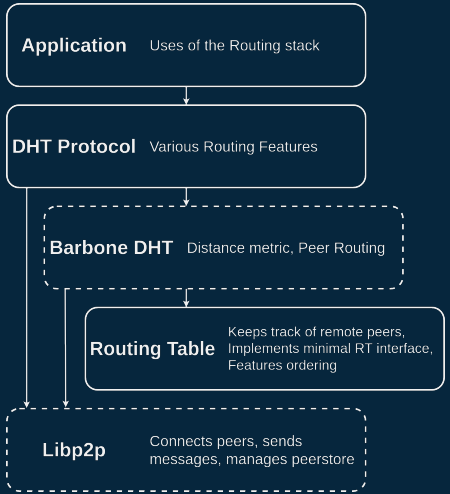
Custom Routing Table Implementations



- Routing Tables can be tailored to fit applications special needs
- Must follow a simple interface with which the Barebone DHT will interact

- Track remote peers' features
- Reachable peers in each bucket
- Implement `FindNClosest(key, feat)`

Example: IPFS Provide Operation



Advertise to the DHT that I provide some content

1. IPFS Impl: asks DHT to provide CID
2. Protocol DHT: find 20 closest peers to CID
3. Barebone DHT: performs recursive query
4. Protocol DHT: allocate Provider Record to the 20 closest peers

Preventing Network Splits

- Record at least `N` peers supporting each feature in each bucket
  - A remote peer can support multiple features
- Decision free to the feature designer, but strongly recommended
- Worst case: the nodes running a common feature are split into multiple partitions, RPC routing won't work as expected
- Can be corrected with `FindFeature`

`FindFeature` **RPC**

- Find other peers running a specific feature
- Can be implemented as a feature
- Upon receiving such a request, a node will return records of peers supporting the requested feature

Bootstrappers nodes

- Any bootstrapper node can be used to join the DHT
- If no provided peers support required features, use `FindFeature` RPC
- Each protocol can alos have its own bootstrap nodes

Main Benefits

- Welcome more applications to the libp2p DHT
- DHT Protocol Agility
- DHT Protocol Interoperability
- Increased Security
- Clear distinction between IPFS and libp2p.

Protocol Upgrade Process

- DHT Protocols are not timeproof
  - New RPCs are added, old RPCs are dropped
  - Basic peer routing should’t change
- Migration transition period
  - New and old features are concurrently supported, before old RPC is dropped
  - For content routing, content must be advertised using both RPCs

Protocol Interoperability

- Libp2p global connectivity + global routability
- Interoperability between multiple libp2p networks using a DHT

- Example: Filecoin could use the DHT to advertise content without having to implement every IPFS DHT RPC.
- Example: Filecoin could have its own DHT protocol optimized for its use, a DHT client could connect to both IPFS and Filecoin DHTs seamlessly
- Example: DHT interoperability between Filecoin, Ethereum, Polkadot etc.

Tradeoffs

- Larger Routing Table
  - Additional storage: $\mathcal{O}(log(n))$
  - Refresh process more expensive: $\mathcal{O}(log(n))$
- Lookup latency stays the same or even decreases in small clusters
- Same routing guarantees in the sub-DHTs as in the current DHT
- Increased security: sybil attacks are more expensive in large networks

Minimal requirements

- Libp2p
- `PeerIDs`
- Standardized Peer Records
- Unified Feature Namespace

Conclusion

- Enables other Libp2p apps to use the DHT
- Brings DHT Protocol Agility & Interoperability
- Changes the Routing Table Selection Process
- Introduces new DHT RPCs as *Features*
- Draws a clear line between IPFS and Libp2p in the DHT

Composable DHT Notion Document (WIP)

Q&A

- All feedback is welcome
- Reach out in `#probe-lab` on Filecoin Slack

Gui Michel (@guissou)

Filecoin Slack