The Lightning Protocol

An Application Design Perspective

BOLT 00: Build Apps

- HTLC is the atom of the App-building Universe
- Channels and the Channel Graph are the environment
- HTLC secret/reveal flow is the ideal. Graph limits you.

BOLT 00: HTLC Lifecycle

1. Funds Locked to Peer
2. Waiting For Peer Response
3. Peer Notification of Failure
4. Peer Notification of Success
5. Chain Notification of Success
6. Chain Notification of Timeout

BOLT 01: The Protocol

- The core protocol is designed to change a lot
- A design constraint: you should stay online
- Forget about other chains, including testnet3

BOLT 02: Peer Protocol

- You’ll need to make channels, even to receive (tricky)
- Channels can close at any time, so get redundancy
- You never know where an HTLC came from

BOLT 03: On-Chain

- The Chain is Lava, it means time-locks and fees
- Small payments aren’t worth going to chain for
- Expect to have time-locked funds

BOLT 04: The Onion

- Sphinx sending is coming, without pre-image proof
- More hops, more time-locks
- Onion APIs are possible

BOLT 05: End States

- Don’t worry about breaches
- Worry about chain fees: be choosy about peers
- Simple backups are easy but not very powerful

BOLT 06: Left Unsaid

- A payment either resolves very quickly, or very slowly
- Payments can easily cost more than on the chain
- Comprehensive backups and shards are a ways off
- Apps can game fees by hiding behind routing nodes

BOLT 07: The Grid

- Stay off the grid with private channels (eventually)
- 90% of the graph is obviously bad
- Graph nodes are your distributed, redundant ISP

BOLT 08: Your Public Key

- Every payment request you make has your key, signature
- Your users have key identities too
- Be careful: custodial users are still possible

BOLT 09: Features so Far

- Reducing channel graph sync requirements: small nodes
- Securing hot wallets with peer enforced output scripts
- No global upgrades yet

BOLT 10: DNS Bootstrap

- You join the network through DNS, like Bitcoin
- DNS is non-judgmental, random
- You want to be judgmental with your actual peers

BOLT 11: Give Me Money

- Huge strings but with lots of useful data for Apps
- Expire timing
- Signed description of Payment
- You can put a fallback chain-address
- Suggest channels to use
- It’s extensible, add your own fields like fiat quote

Read the Docs

- https://github.com/lightningnetwork/lightning-rfc
- https://lists.linuxfoundation.org/pipermail/lightning-dev/
- https://github.com/alexbosworth/ln-service