Improving Routing in the Lightning Network with Trampoline Payments

Scaling Bitcoin 2019

Bastien Teinturier (ACINQ)
@acinq_co @realtbast
The Lightning Network, a Payment Channels Network

You can pay anyone you can find a route to!
Payment Routing: Fees and Lock Time

Intermediate channels advertise a fee rate:

- base fee
- proportional fee

And a lock time delta:

- to protect their funds
- and avoid locking capacity for too long

Gossiped via staggered broadcast.
Payment Routing Features

Powerful features

- **Optimal path (source routing)**
- **Trustless (HTLCs)**
- **Private (onion encryption)**

But at a cost…

- **Bandwidth**
- **Memory**
- **CPU**
- **Which impacts mobile UX**

*Does not know* its position in the route i.e
- Does not know who the sender is
- Does not know who the final recipient is
Lightning Payment Triangle of Success ©

- Privacy: we’re here!
- Route cost optimum
- Performance: this side looks interesting too...
- We really don’t want to be there...
Let a billion payment channels bloom...

A phone isn’t a server (duh):
- not always online
- unreliable connections
- limited performance
- handy payment method

What if?
- sync smaller graph
- and special nodes
- trade fees for performance
Partial source routing:

- select trampoline nodes
- route to first one

Routing between trampoline nodes is deferred to the trampoline nodes themselves.

Powered by variable-length onions (onion nesting FTW).
Gossip Filters

Reduce bandwidth usage:

- *channel_update* filters
  - n-radius
  - channel capacity
- *node_update* filters
  - fee rate
  - lock time
  - probabilistic (distance to block hash?)

Applied *before* forwarding gossip.

Without BIP 37 drawbacks!
Fee Estimation

Trampoline nodes broadcast aggregated trampoline fees and lock time (node_update).

Mobile nodes may build a reputation score DB to prune unreliable trampoline nodes.

Custom heuristics may be added depending on node connectivity and performance.

\[ f = 5, \% = 0.5, \Delta = 12 \]
\[ f = 3, \% = 0.2, \Delta = 6 \]
\[ f = 2, \% = 0.05, \Delta = 5 \]
\[ f = 10, \% = 0.01, \Delta = 3 \]
\[ f = 12, \% = 0.5, \Delta = 15 \]
AMP 🎈 Trampoline

#reckless
To Infinity and Beyond

Incremental change to payment infrastructure.

Easy to rollout.

Coming to eclair soon.

Incremental step towards hierarchical systems.

Feedback wanted!
https://github.com/lightningnetwork/lightning-rfc/pull/654
Questions?