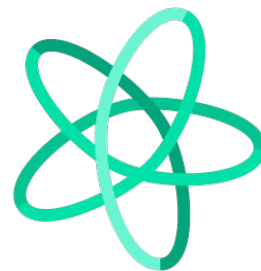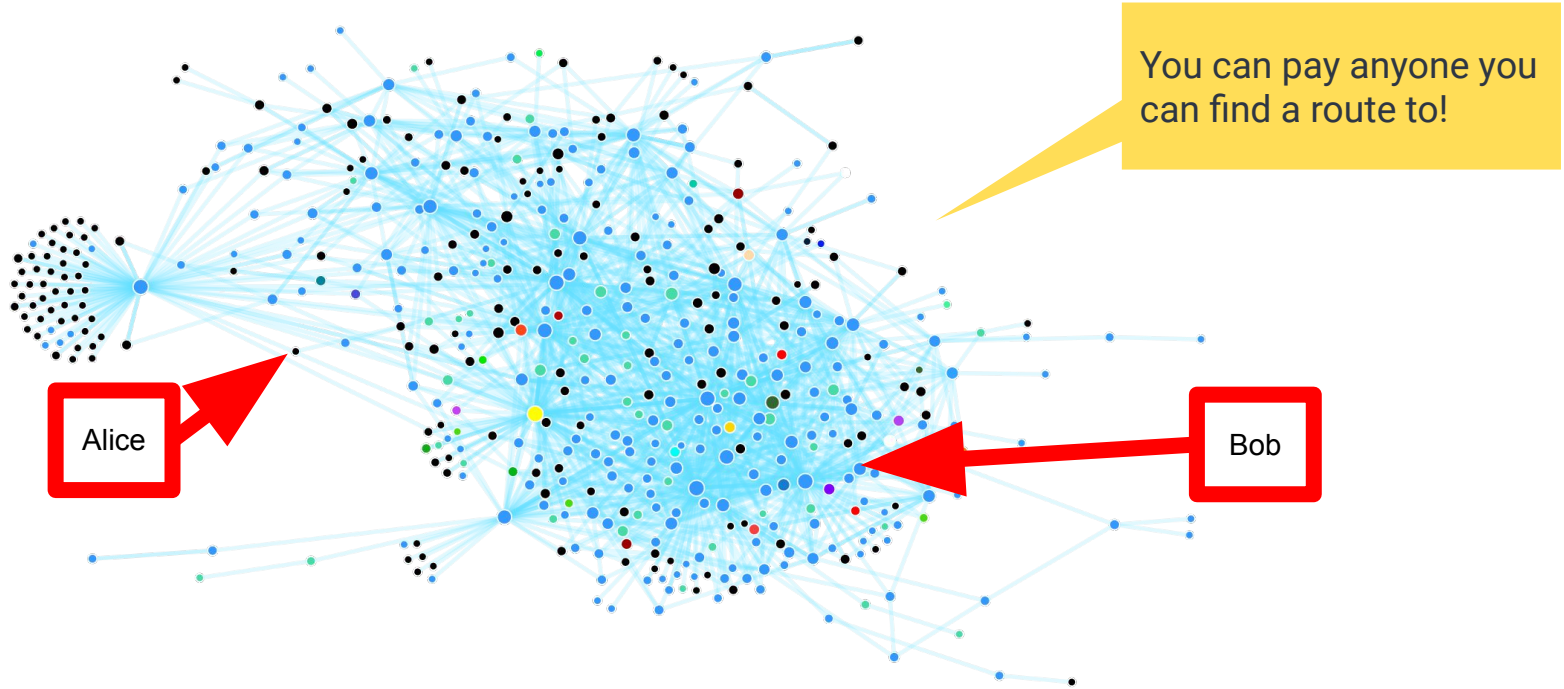# Improving Routing in the Lightning Network with Trampoline Payments

Scaling Bitcoin 2019

Bastien Teinturier (ACINQ)
@acinq_co @realtbast

The Lightning Network, a Payment Channels Network
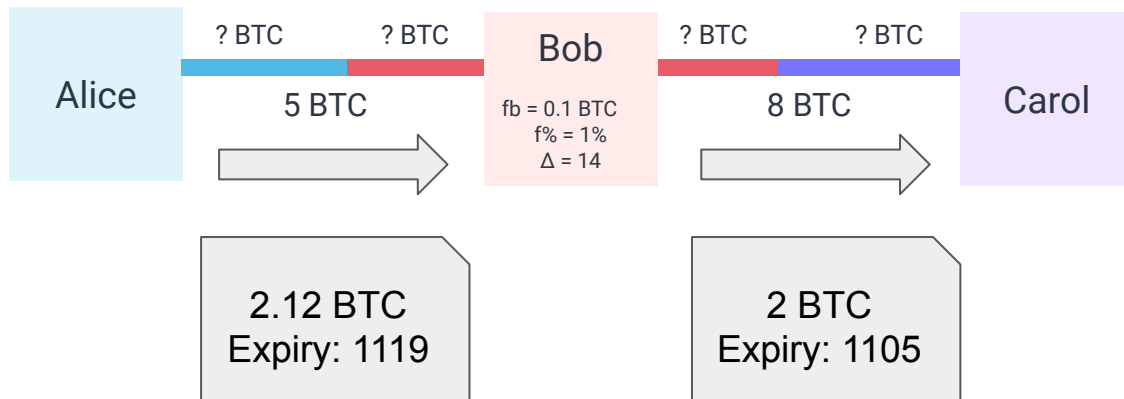
Payment Routing: Fees and Lock Time

Intermediate channels advertise a fee rate:

- base fee
- proportional fee

And a lock time delta:

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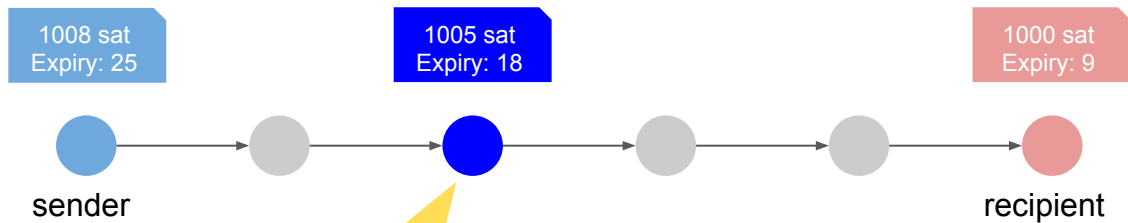- and avoid locking capacity for too long

Gossiped via staggered broadcast.

Payment Routing Features

Powerful features

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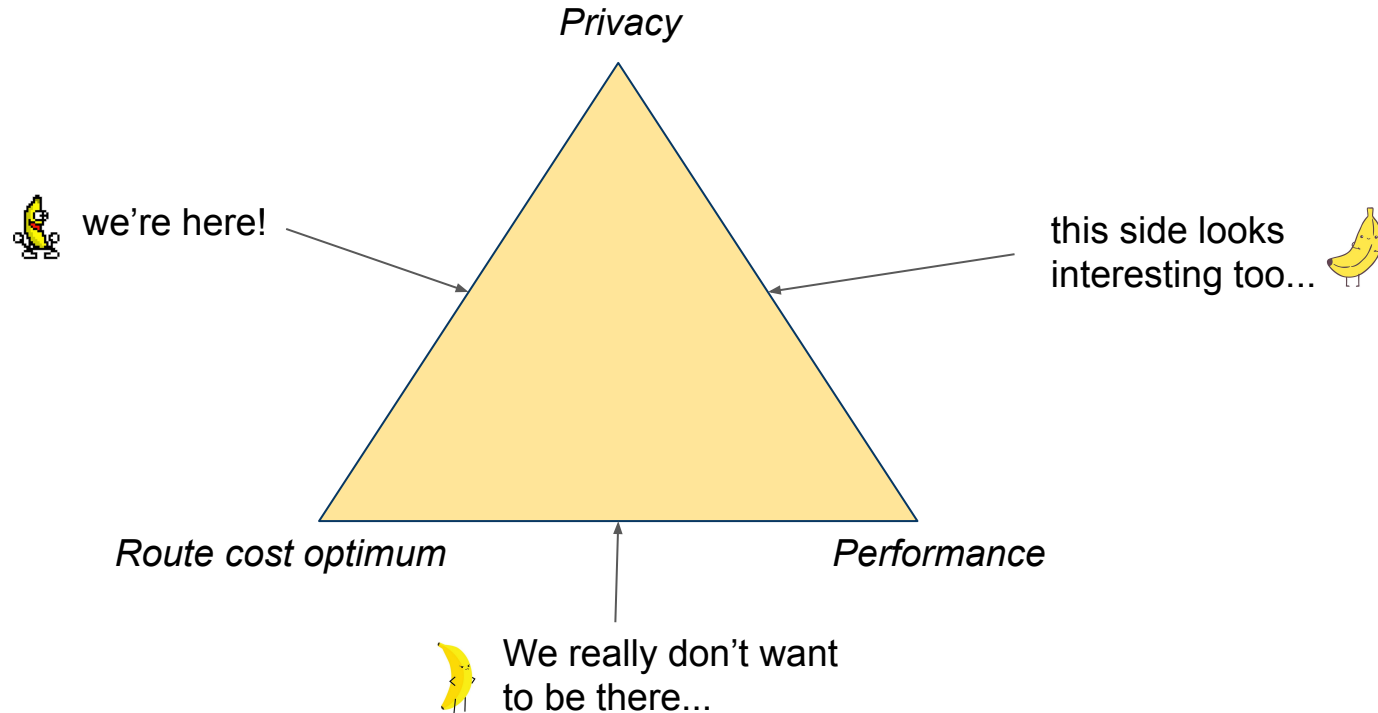- *Trustless (HTLCs)*
- *Private (onion encryption)*

But at a cost…

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

1008 sat
Expiry: 25

1005 sat
Expiry: 18

1000 sat
Expiry: 9

sender

recipient

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!

this side looks
interesting too...

Route cost optimum

Performance

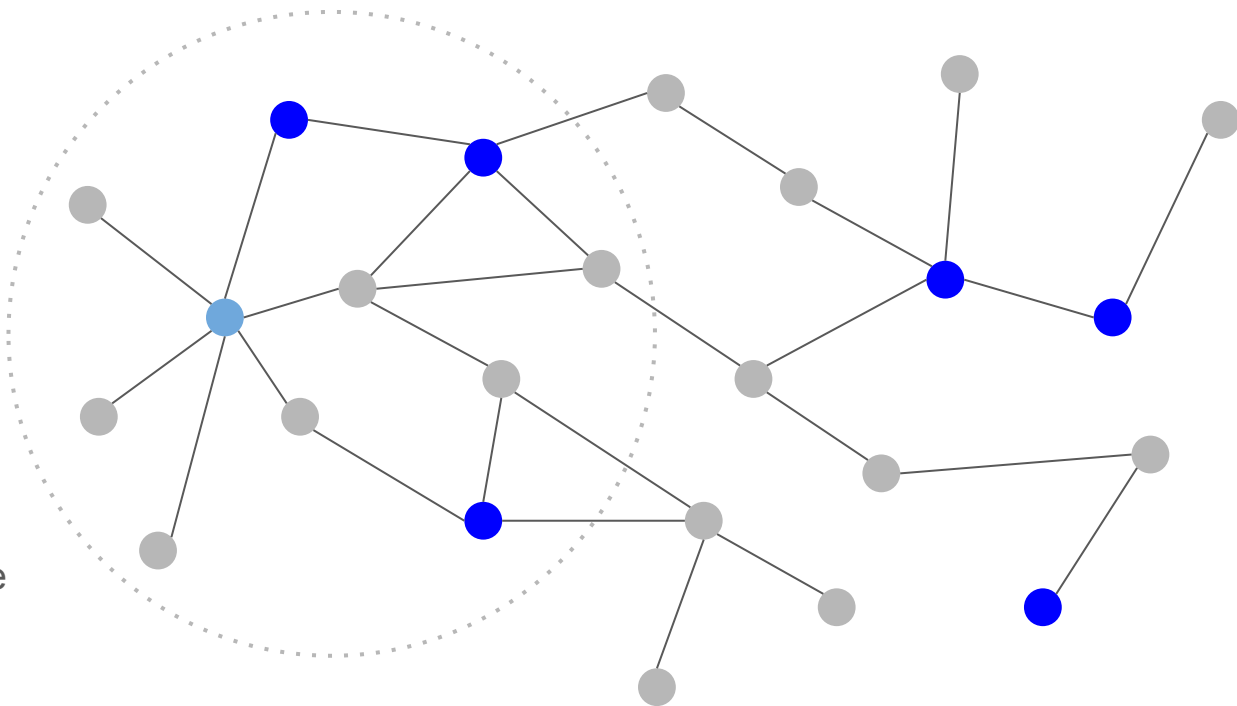We really don't want
to be there...

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
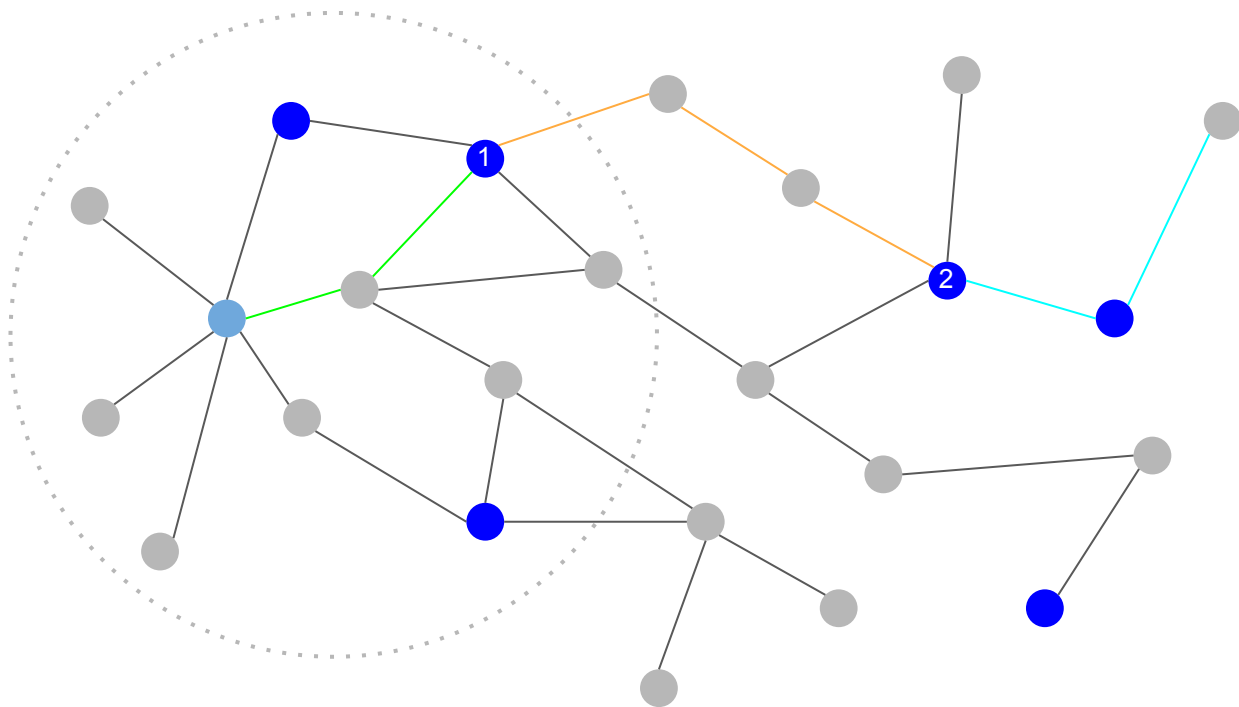- trade fees for performance

Trampoline Routing

Partial source routing:

- select trampoline nodes
- route to first one

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

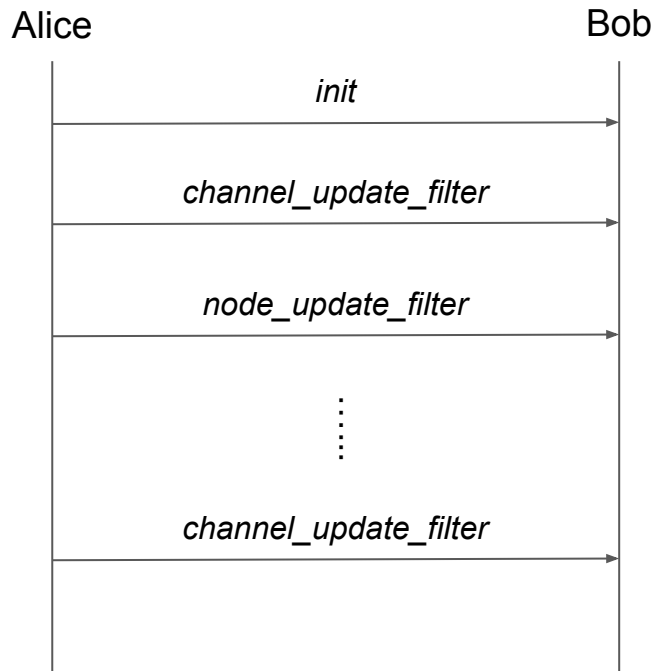Powered by variable-length onions (onion nesting FTW).

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!

Alice   Bob

init

channel_update_filter

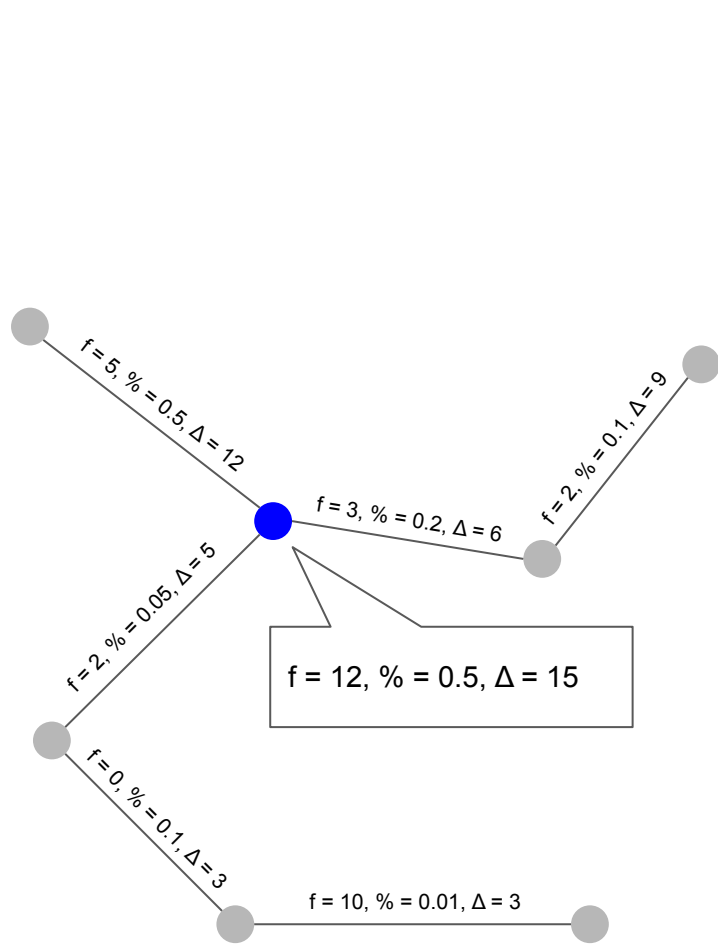*node_update_filter*
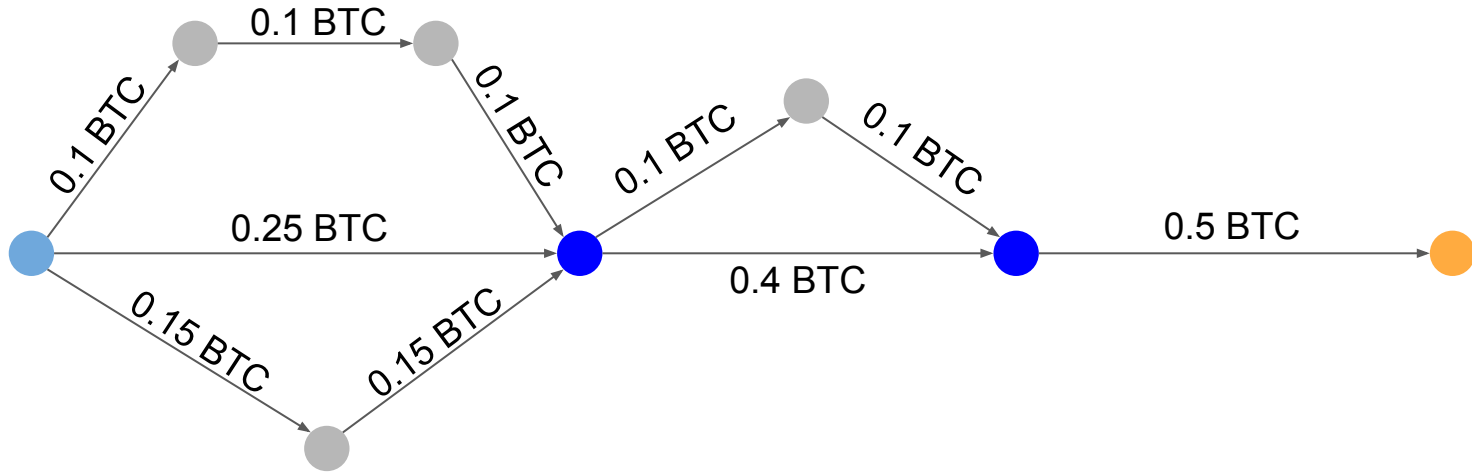
⋮

channel_update_filter

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.

AMP 💙 Trampoline

#reckless

To Infinity and Beyond

Incremental change to payment infrastructure.
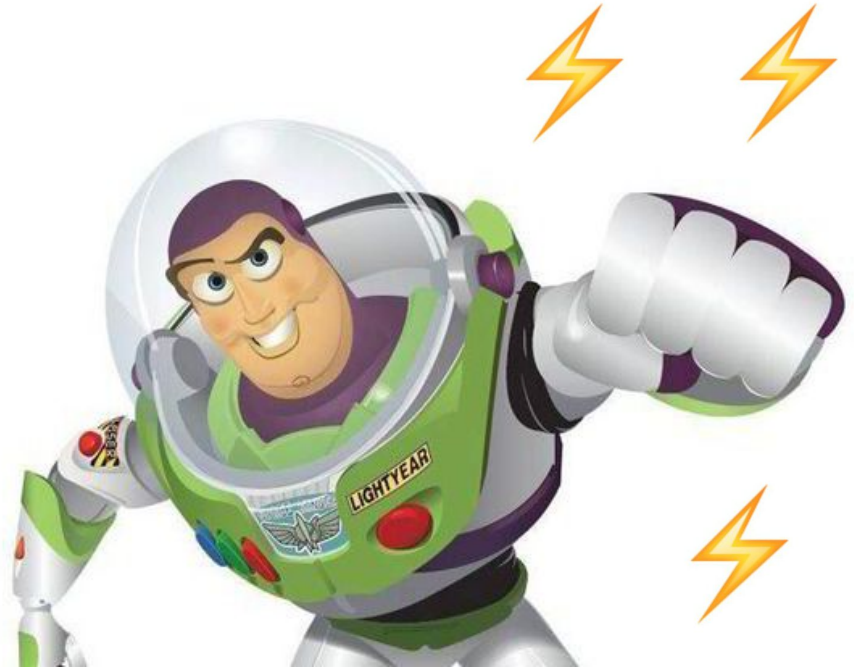
Easy to rollout.

Coming to eclair soon.

Incremental step towards hierarchical systems.

Feedback wanted!
https://github.com/lightningnetwork/lightning-rfc/pull/654
https://lists.linuxfoundation.org/pipermail/lightning-dev/2019-August/002100.html

Questions?