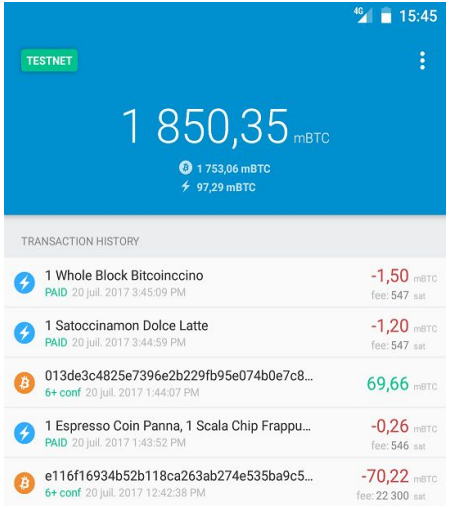
Lightweight Clients

Chaincode LN Residency - NY 2019

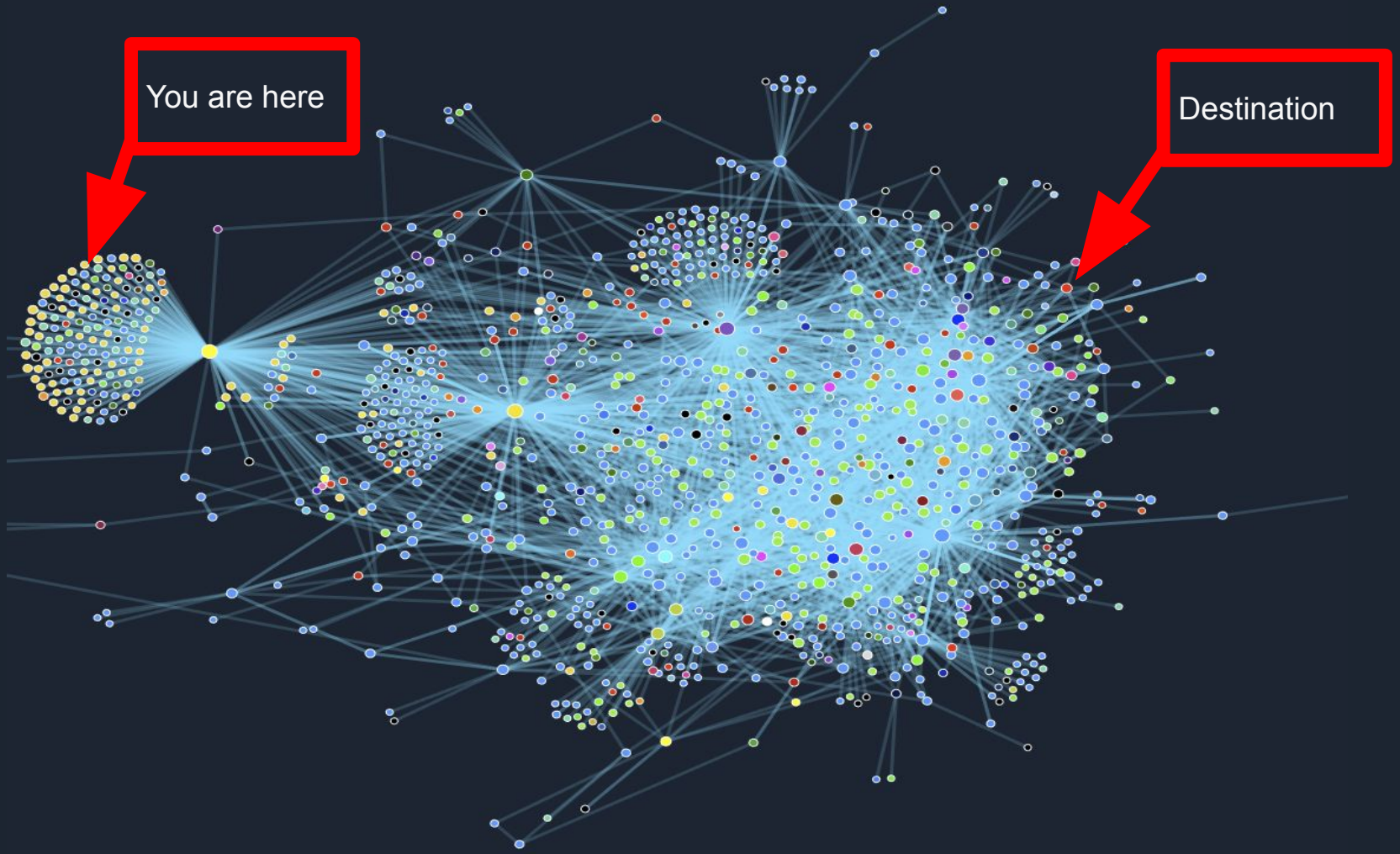
Fabrice Drouin <fabrice.drouin@acinq.fr> - https://acinq.co/

How do Payments work ?

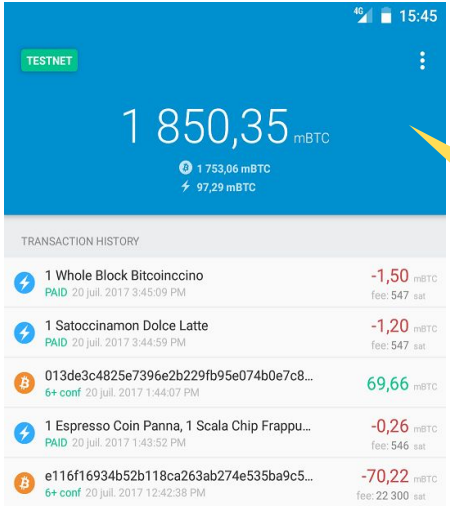


Scan Payment Request

Extract amount, destination, expiry and payment hash

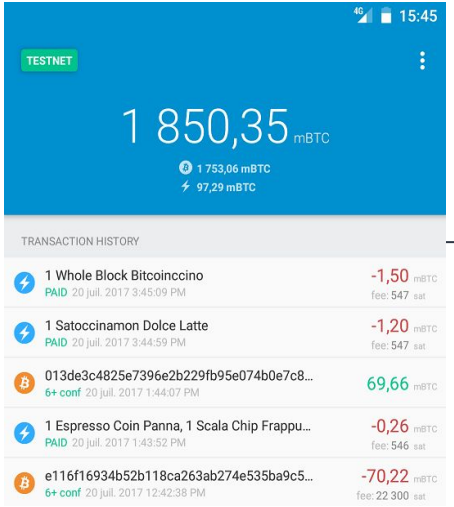

You are here
Destination

How do Payments work ?

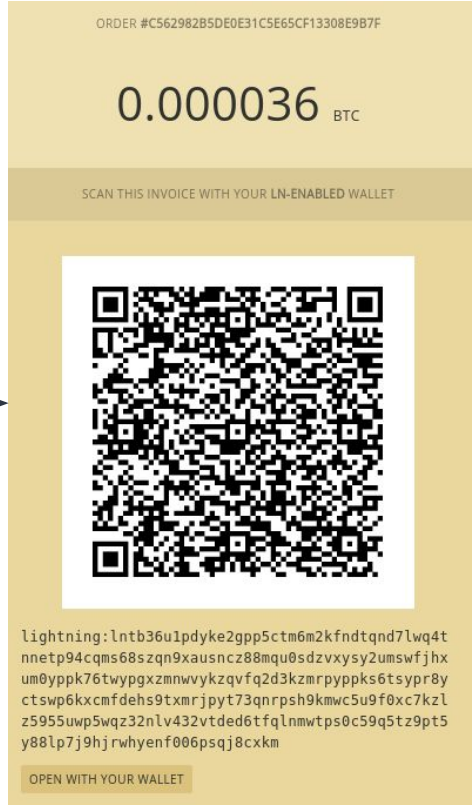


Computes route to destination and creates an onion packet

How do Payments work ?

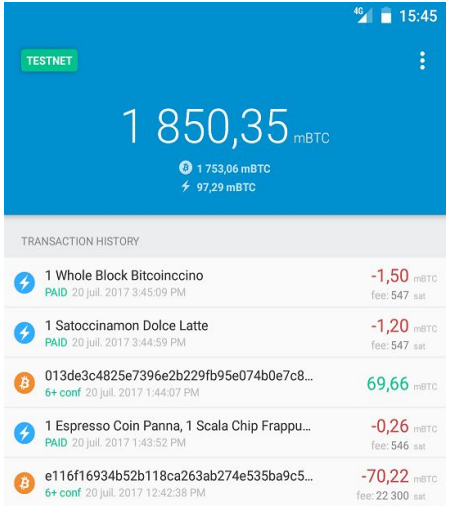


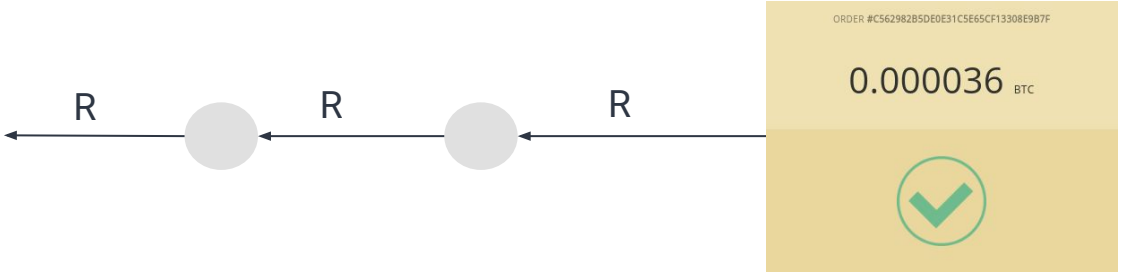
HTLC → HTLC → HTLC →



Payment hash is used as payment identifier

How do Payments work ?





TL;DR

A Lightning Node:

- Creates, signs and sends bitcoin transactions
- Monitors the blockchain to detect when funding transactions are spent
- Maintains a routing table that it uses to compute payment routes

Building a mobile Lightning Node

Implementing a mobile Bitcoin Wallet

- Cannot store the whole bitcoin blockchain: we need a “light” wallet
  - Bloom Filters (BIP37) ?
  - Neutrino (BIP157/158) ?
  - Use an API ?
  - Roll our own servers ?
  - Use Electrum servers ?
- We chose to build a simple bitcoin wallet that uses Electrum servers
  - Not too heavy development-wise
  - Interests aligned with Electrum
- We are following BIP157/158 and will probably develop our own “neutrino” wallet

Monitoring the Blockchain

- Mobile nodes are often offline
- If you’re offline and your peer publishes an old state, you may miss the “penalty window” and lose money
- Use very long penalty time-outs ?
- Delegate Blockchain monitoring to a 3rd party (“watchtowers”) ?

Monitoring the Blockchain

- Alice opens a 100 mbtc channel to Bob, and starts buying things
  - State #1: 100 to Alice, 0 to Bob
  - State #2: 90 to Alice, 10 to Bob
  - State #3: 85 to Alice, 15 to Bob
  - ….
- If Alice only sends money and never receives, Bob’s balance keeps going up
  - Why would Bob publish an old state ?
  - An if he did, Alice would get more money than what she’s supposed to have
- We chose to start with a mobile node than can only send, not receive
  - No need to worry about being online for too long
  - Makes sense from a UX point of view
- We then added the option to receive money
  - once we had a good way of providing inbound liquidity

Lightning Watchtowers

The ideal watchtower should:

- Watch the blockchain for you
- Monitor your channels
- React when your peers try to cheat by publishing a penalty transaction
- Learn nothing about your payments

Lightning Watchtowers

One of the first watchtower designs:

- Nodes send watchtower one penalty tx for each commit tx
- Penalty tx is encrypted with the last 16 bytes of the commit tx id
- Watchtowers only know the first 16 bytes of the commit tx id

Nice !

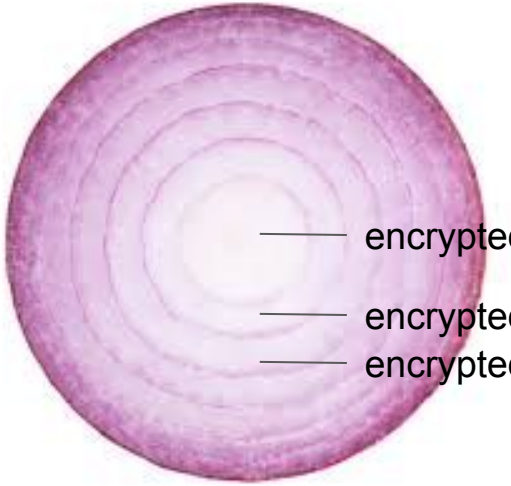
- Watchtowers don’t know which channels they’re watching
- If no one tries to cheat, they learn absolutely nothing

But:

- Need to store an ever increasing amount of opaque data….

Routing Table Management

- Source routing + onion packets
  - Source computes the route to the destination
  - Messages wrapped in an onion-like packet
  - Intermediary node don't know the final destination
- Nodes need to have an up-to-date routing table
  - Easy when you're online most of the time
  - Hard when you're offline most of the time
    - Need to know about new channels
    - Need to know about channels that have been closed



"First payment is slow" issue

Routing Table Management

- Routing Table Sync
  - Node starts and needs to update its routing table
- 1st version: download everything every time you start
  - Very inefficient on mobile devices
  - Was beginning to be a problem even for server nodes
- 2nd version: download all channel ids, and then ask for what you need
  - 8 bytes per channel id: 80 Kb for 10 000 channels!
- 3rd version: smarter queries
  - filter by timestamp and content
- Could be further improved with set-reconciliation techniques (IBLT, minisketch…)

Routing Table Management

- Delegate route computation to a central server ?
  - Privacy issue: defeats the point of using source routing
- Use a external scheme for routing table syncing ?
  - Requires some level of trust
  - But better than using a central server to compute routes
- Trampoline payments !
  - Allow intermediate nodes to compute a subpath

Backups

What if I lose my phone ?

- Restoring your wallet seed will get you your onchain funds back
- But not your offchain LN funds...

VS

What if I restore an old backup ?

- You would publish an old state and potentially lose all your funds...

Backups

We added encrypted backups to our testnet wallet

- The backup file is encrypted with a key derived from the wallet seed
  - 1 seed = 1 backup
- Stored on the user's Google Drive Application folder
- Updated when the state of channels change (~6 times for 1 payment)
- Upon seed restore, the wallet can restore the backup file for this seed
- If old state, rely on dataloss protection

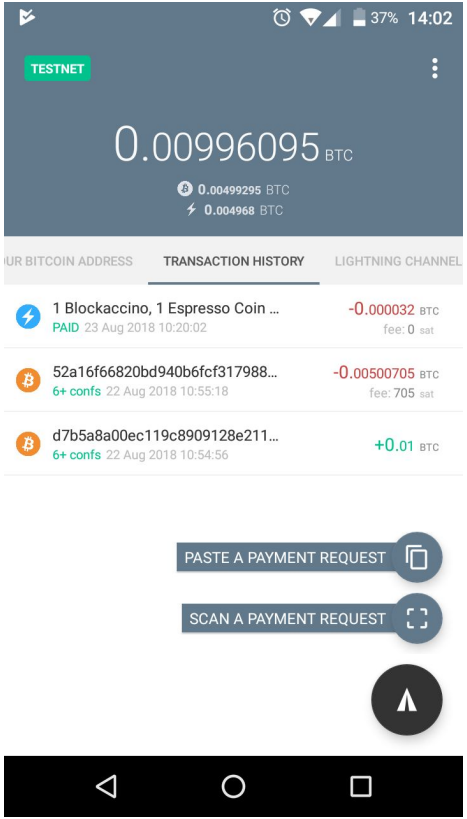
- ⚠️ not a synchronization tool!

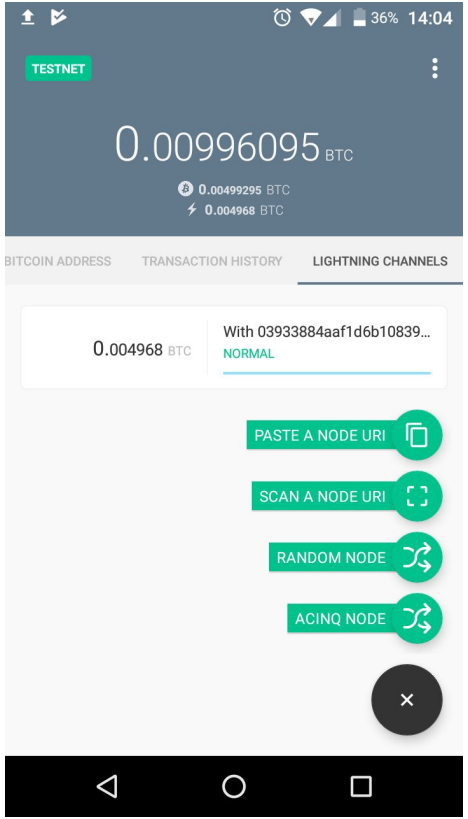
Implementation issues

- Android vs iOS
  - No good cross-platform dev tools for native apps
  - No production-ready JVM on iOS
    - Several issues (JIT for example) -> compile to native code ?
  - LN on iOS: develop a new LN core library from scratch, or use experimental tools

- Android specific issue
  - JVM is stuck at 1.7, which is becoming obsolete
    - Many libraries require 1.8 or higher, older versions not maintained
    - Special version of eclair-core for android

UX Issues


37% 14:02
TESTNET
0.00996095 BTC
0.00499295 BTC
0.004968 BTC
UR BITCOIN ADDRESS
TRANSACTION HISTORY
LIGHTNING CHANNEL
1 Blockaccino, 1 Espresso Coin ...
PAID 23 Aug 2018 10:20:02
-0.000032 BTC
fee: 0 sat
52a16f66820bd940b6fcf317988...
6+ confs 22 Aug 2018 10:55:18
-0.00500705 BTC
fee: 705 sat
d7b5a8a00ec119c8909128e211...
6+ confs 22 Aug 2018 10:54:56
+0.01 BTC
PASTE A PAYMENT REQUEST
SCAN A PAYMENT REQUEST


36% 14:04
TESTNET
0.00996095 BTC
0.00499295 BTC
0.004968 BTC
BITCOIN ADDRESS
TRANSACTION HISTORY
LIGHTNING CHANNELS
0.004968 BTC
With 03933884aaf1d6b10839...
NORMAL
PASTE A NODE URI
SCAN A NODE URI
RANDOM NODE
ACINQ NODE

Bonus Slides

Routing Table Management

Routing Table Management

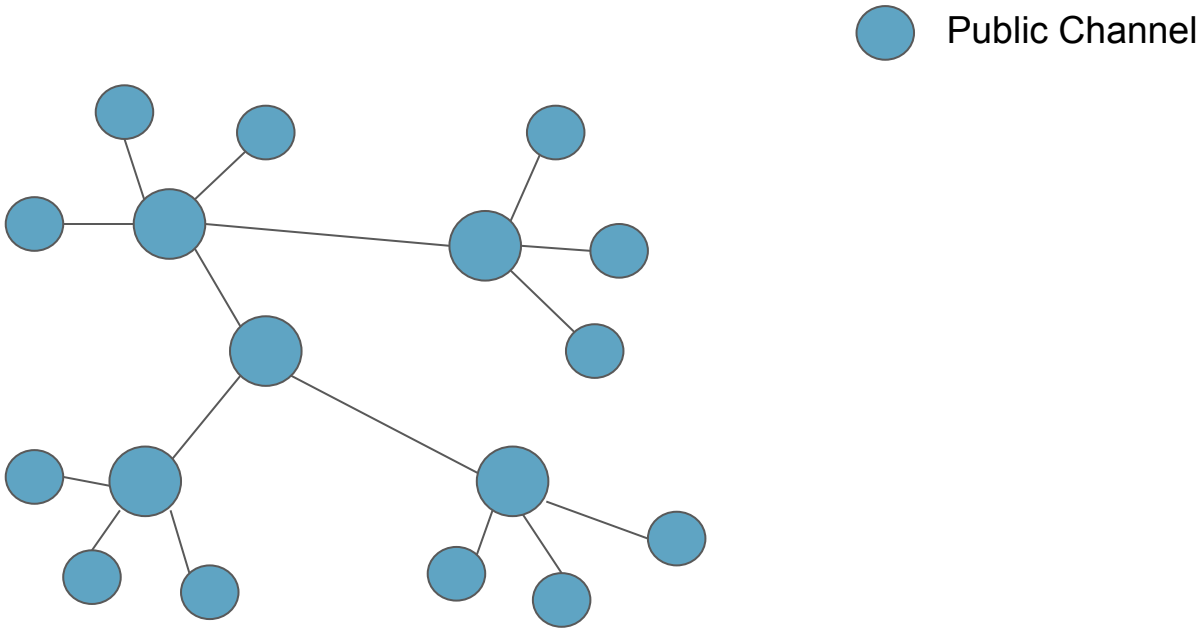
Only channels that relay payment have to be announced !

- Not even all of them, you can have several channels between 2 peers, announce only one of them but still use the others (“channel override”)

Terminal nodes (that send or receive payments but do not relay them) don’t need to be added to the routing table

- Mobile nodes, “personal” nodes on laptops … that are often offline
- There could be **millions** of them, it would have no impact on the size of the routing table
- There are many more channels than what you can see on LN explorers !
- You can also build a “hidden” network of unannounced channels

What you see



What you don’t see