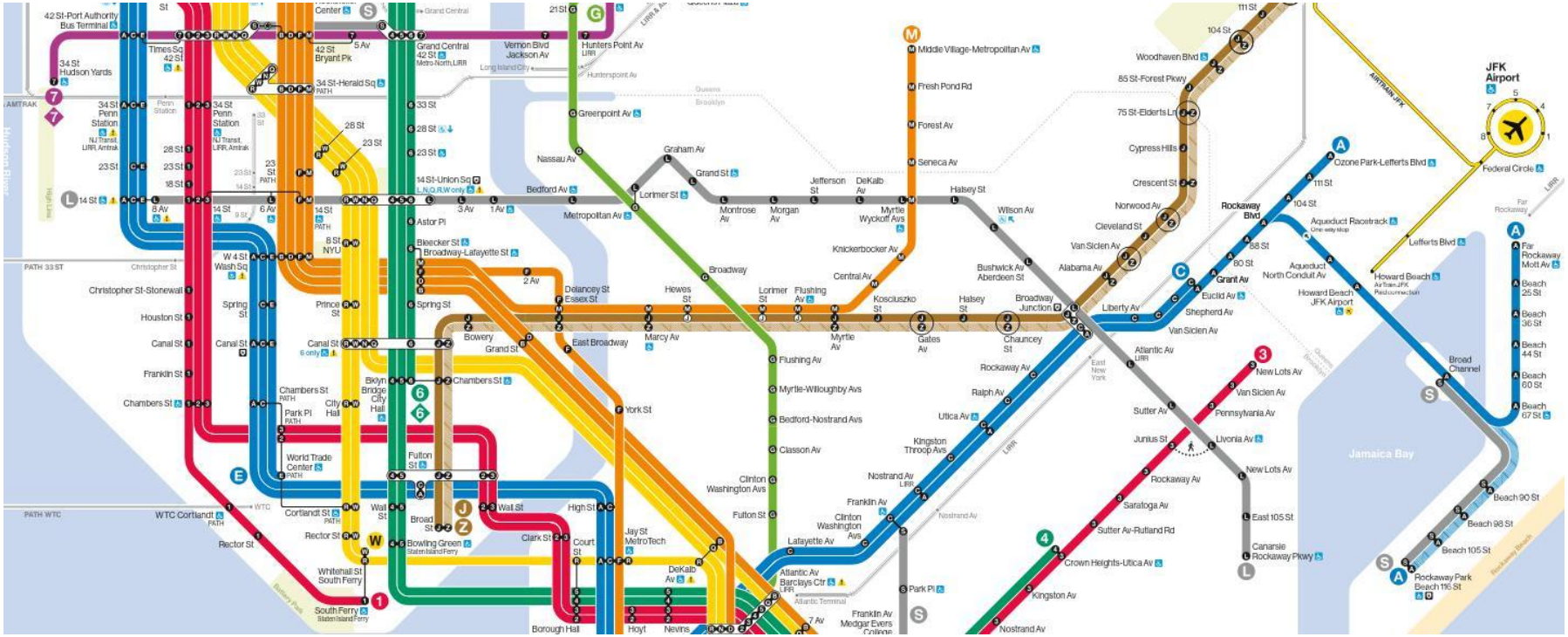
Spilling the 🧋: Gossip (v2) and Minisketch

@jonhbit / jharveyb

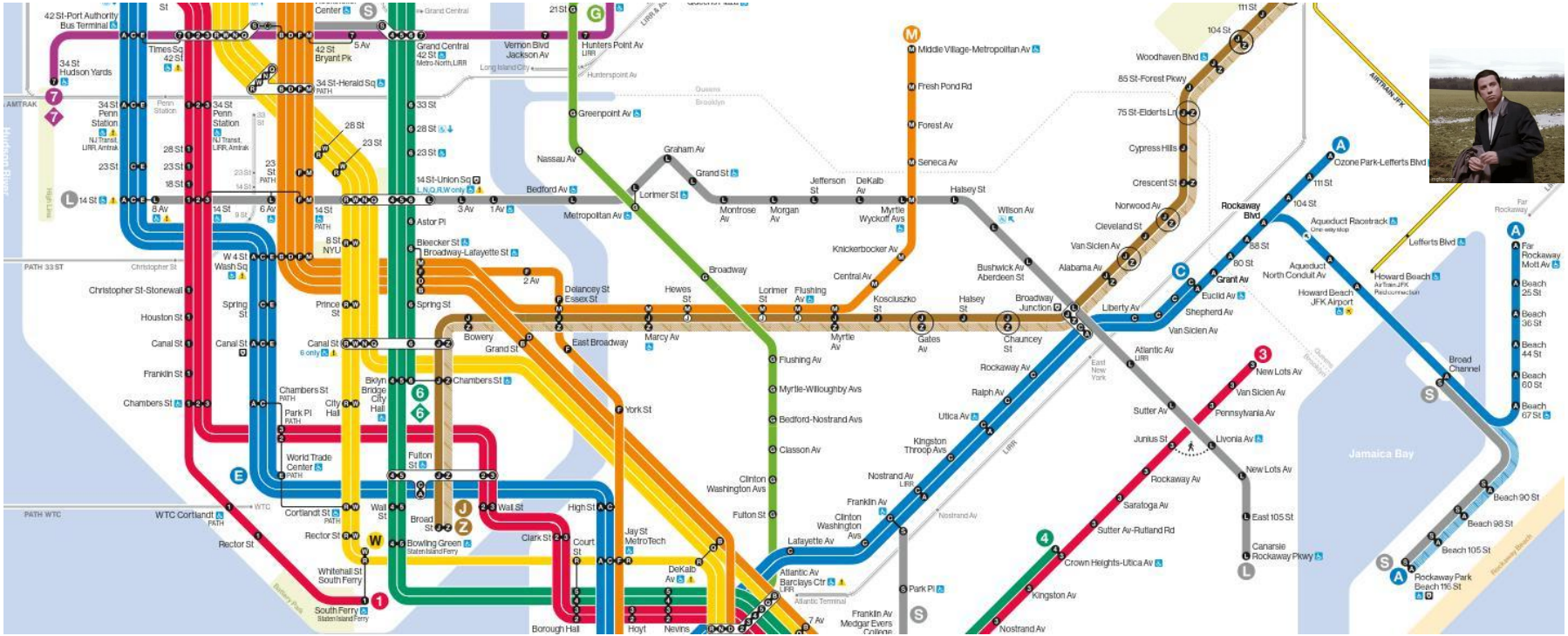
Overview

- Gossip: ELI18
- Previous Work & Defaults
- New data!
- Minisketch & Gossip v2

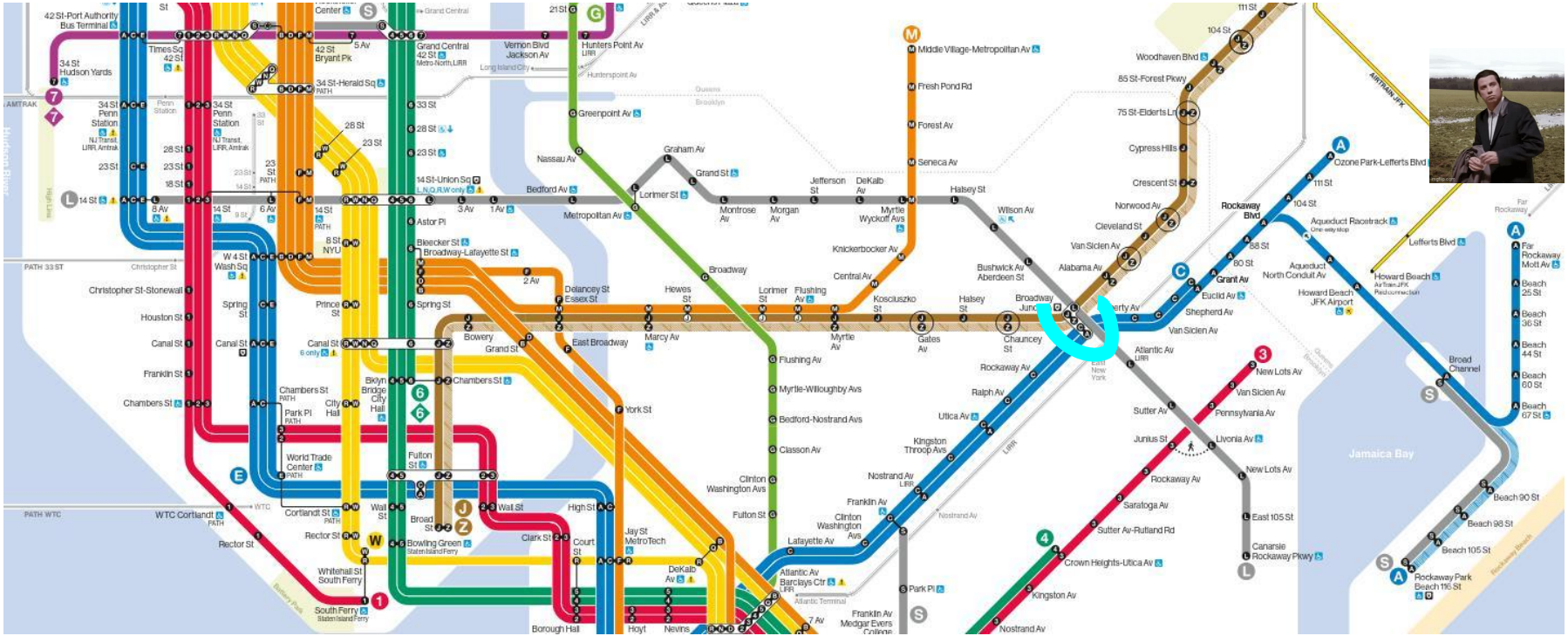
Gossip: ELI18



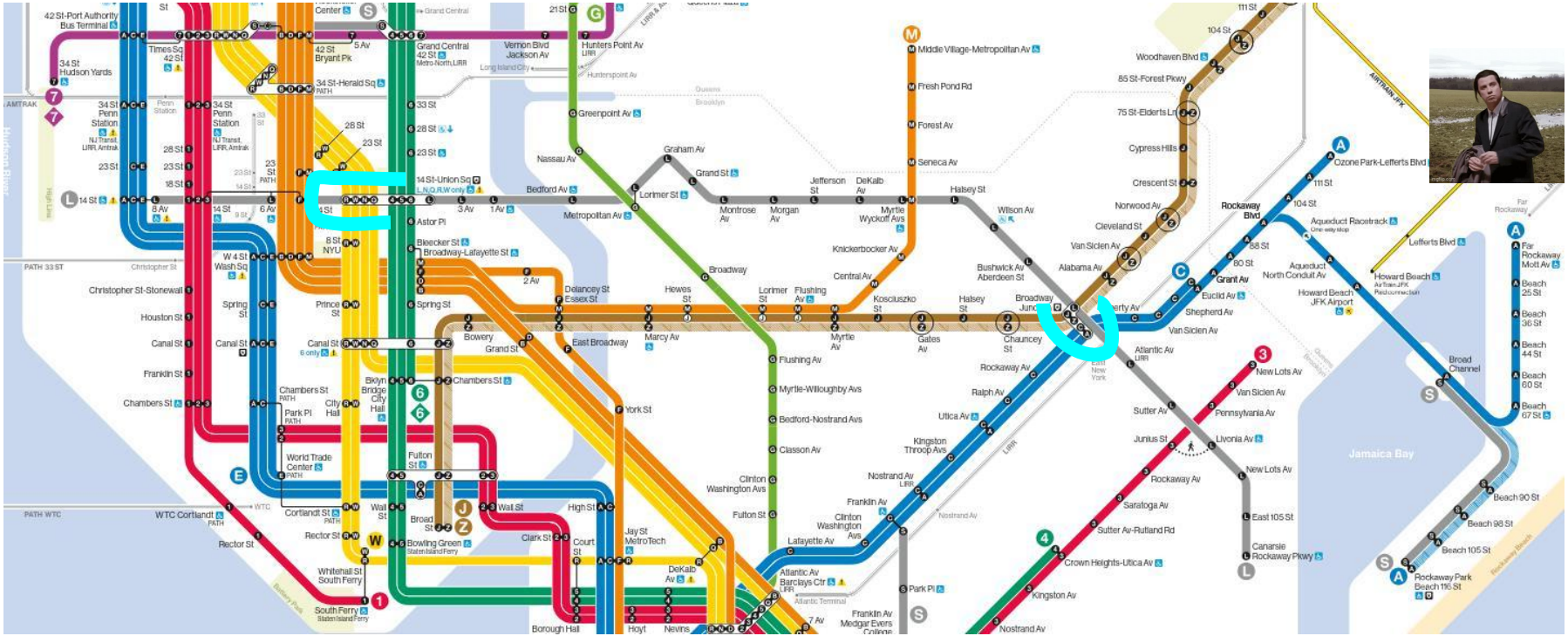
Gossip: ELI18



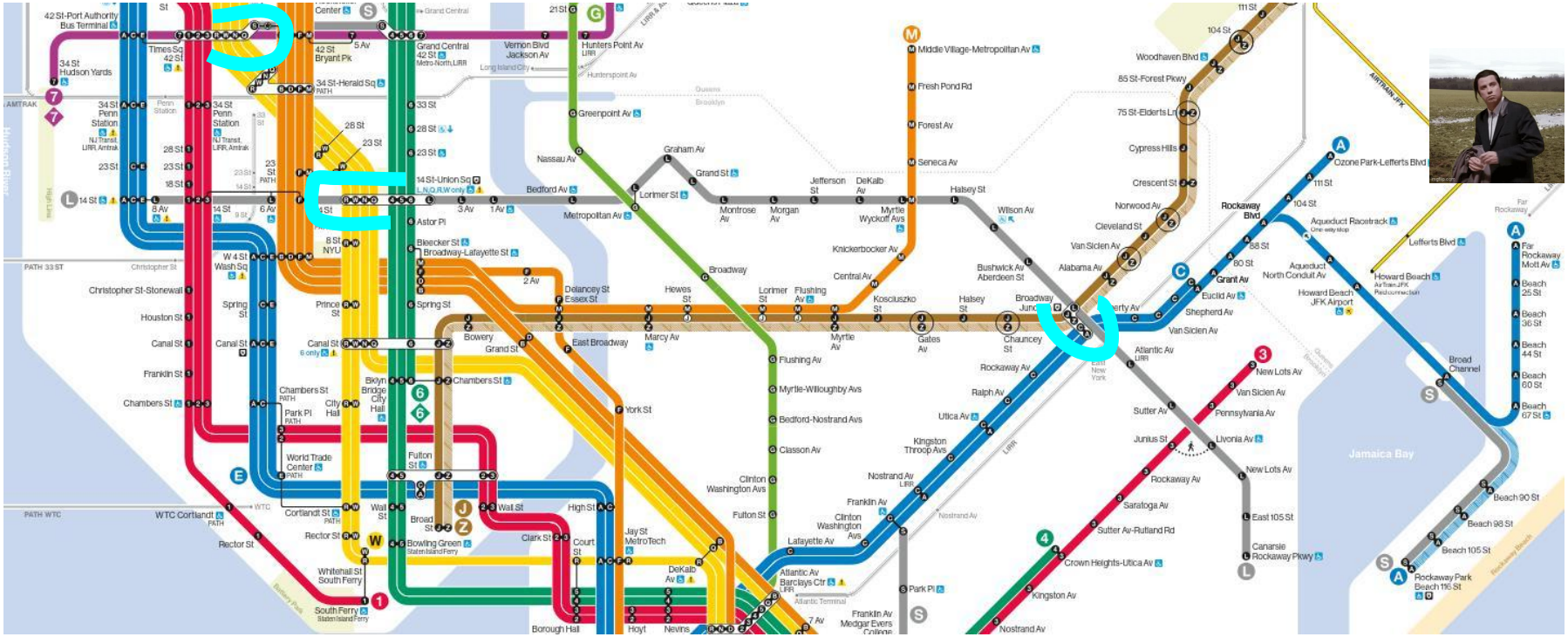
Gossip: ELI18



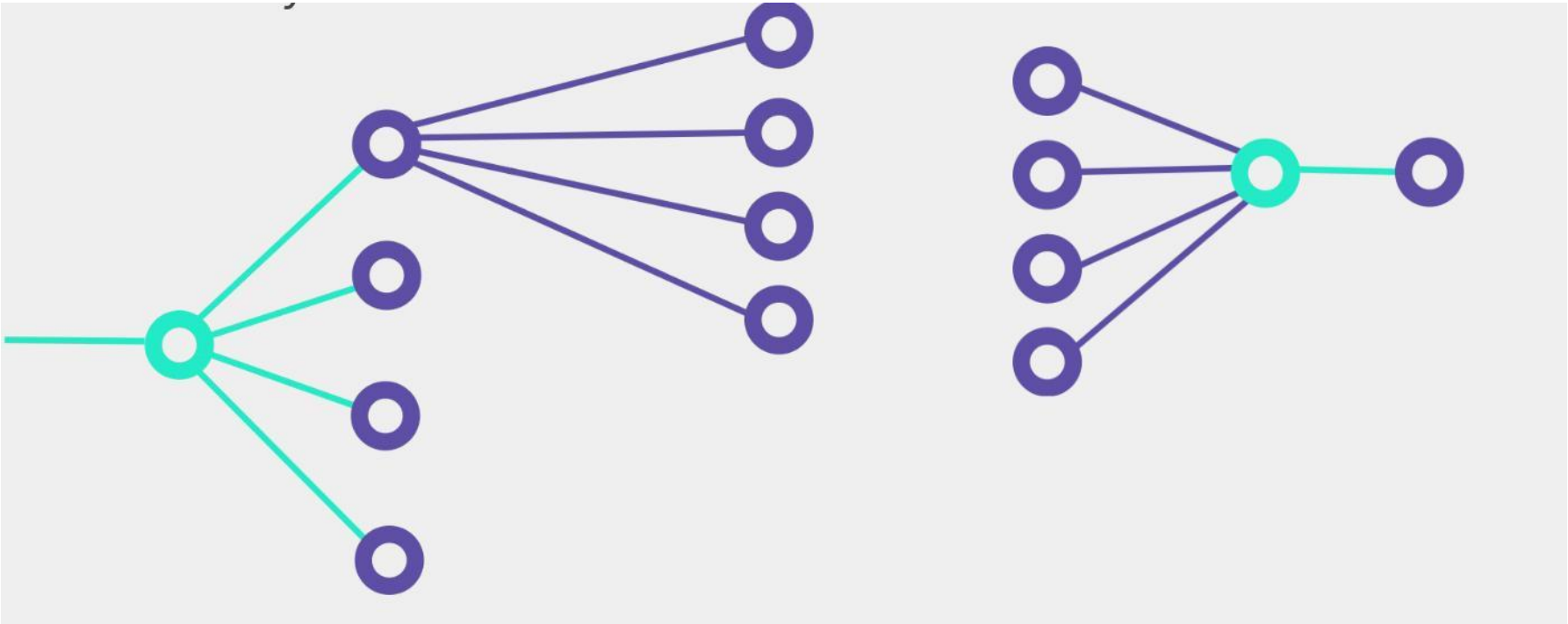
Gossip: ELI18



Gossip: ELI18



Gossip: Flooding



Gossip: Does it work?

Gossip: Does it work?

```
625    // The lightning network's gossip sync system is completely broken in numerous ways.
626    //
627    // Given no broadly-available set-reconciliation protocol, the only reasonable approach is
628    // to do a full sync from the first few peers we connect to, and then receive gossip
629    // updates from all our peers normally.
630    //
631    // Originally, we could simply tell a peer to dump us the entire gossip table on startup,
632    // wasting lots of bandwidth but ensuring we have the full network graph. After the initial
633    // dump peers would always send gossip and we'd stay up-to-date with whatever our peer has
634    // seen.
635    //
636    // In order to reduce the bandwidth waste, "gossip queries" were introduced, allowing you
637    // to ask for the SCIDs of all channels in your peer's routing graph, and then only request
638    // channel data which you are missing. Except there was no way at all to identify which
639    // `channel_update`s you were missing, so you still had to request everything, just in a
640    // very complicated way with some queries instead of just getting the dump.
641    //
```

Gossip: Does it work?

- “I think our channel updates aren’t propagating”
- “DoS-like” behavior with (serving) gossip queries
- “Inbound gossip volume caused performance issues; we turned it off”
- “Our channel_announcement never propagated to our other nodes”
- Cross-implementation issues handling certain fields

“Measure Once, Cut Twice”

Previous Work

- https://github.com/lnresearch/topology
  - Archives of gossip from multiple Core Lightning nodes, 2020-2023
- https://github.com/lnresearch/gossip
  - New gossip archives, starting from early 2025

- Total traffic, graph snapshots, rate-of-change of the graph

Previous Work

On the Routing Convergence Delay in the Lightning Network

Niklas Gögge[1], Elias Rohrer[2], and Florian Tschorsch[2]

[1] Distributed Security Infrastructures
Technical University of Berlin
n.goegge@campus.tu-berlin.de
[2] Distributed Security Infrastructures
Technical University of Berlin
{elias.rohrer, florian.tschorsch}@tu-berlin.de

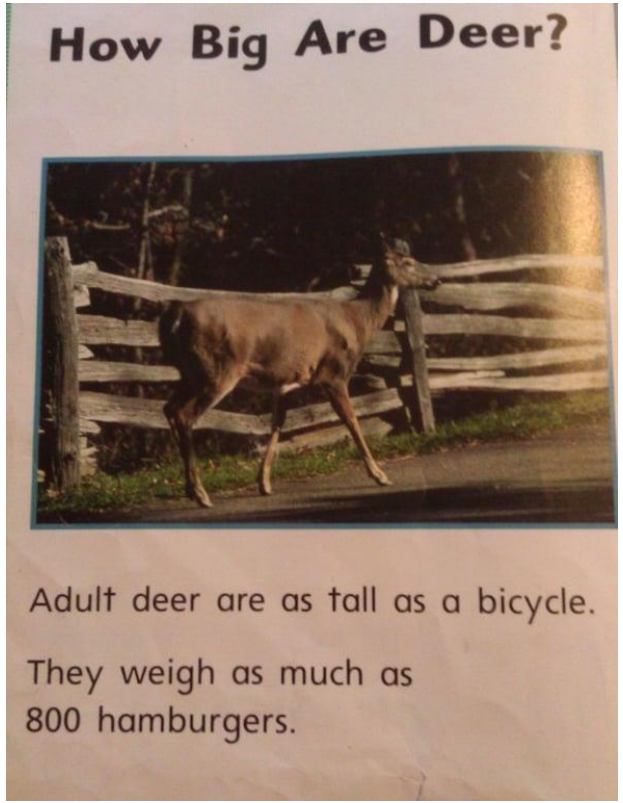
Previous Work

On the Routing Convergence Delay in the Lightning Network

Niklas Gögge[1], Elias Rohrer[2], and Florian Tschorsch[2]

[1] Distributed Security Infrastructures
Technical University of Berlin
n.goegge@campus.tu-berlin.de
[2] Distributed Security Infrastructures
Technical University of Berlin
{elias.rohrer, florian.tschorsch}@tu-berlin.de

Convergence Delay? Metrics?



Metrics

- Convergence Delay: Time for **X%** of nodes to receive a gossip message
- Size: Bytes needed for unique messages
- Bandwidth: Bytes sent or received, per unit time
- Frequency: How often we see a channel_update or node_announcement for a particular SCID or node

Methodology

- Fork of ldk-node; export all received gossip messages, don’t forward anything
- Random set of clearnet nodes
- Store the sending peer, time of receipt

Diffing: 2021->25

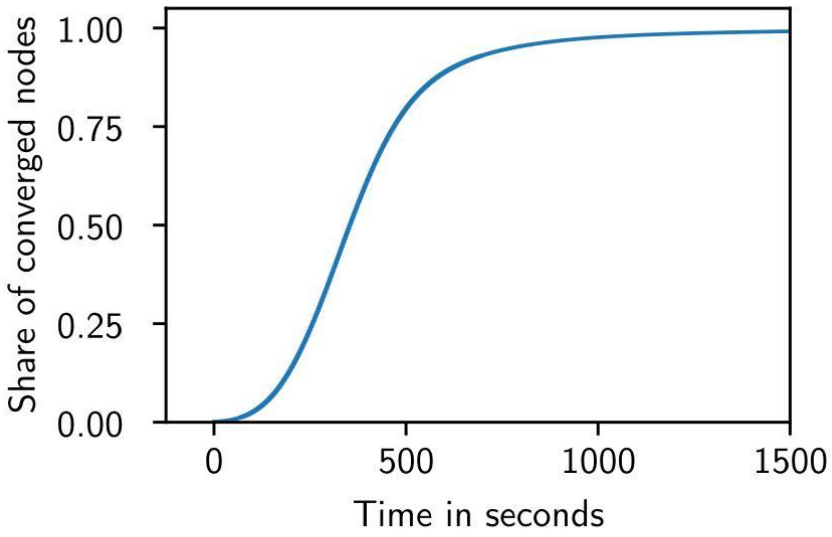
- Core Lightning default # of gossip syncers, 5 -> 10
- LND & Core Lightning both have outbound B/W limits of 1 MB/s
- Rotate peers based on “quality” of received gossip
- Rust-lightning will request last 2 weeks from the first 5 peers
- Eclair syncs from 5 peers, ranked by channel capacity; no randos

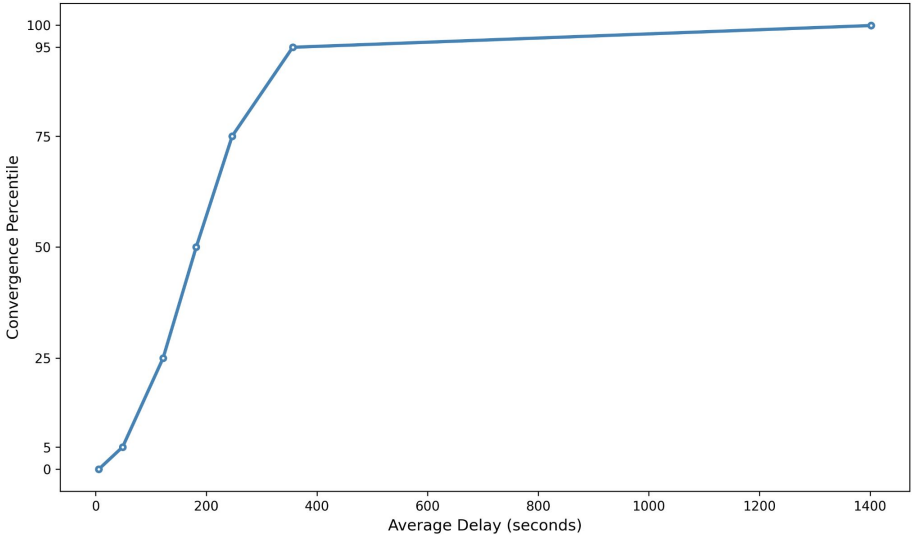
Diffing: 2021->25

- ~17000 nodes, ~78000 channels
- ~116.5 unique msgs. / minute
- 94% channel_update
- 5% node_announcment
- 1% channel_announcement

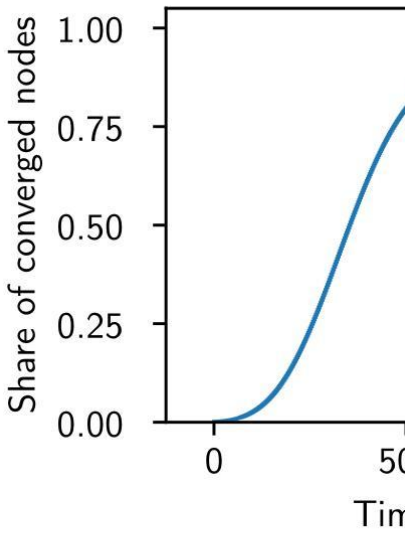
- ~15000 nodes, ~45000 channels
- ~295.3 unique msgs. / minute
- 60% channel_update
- 30% node_announcement **(!)**
- 10% channel_announcement

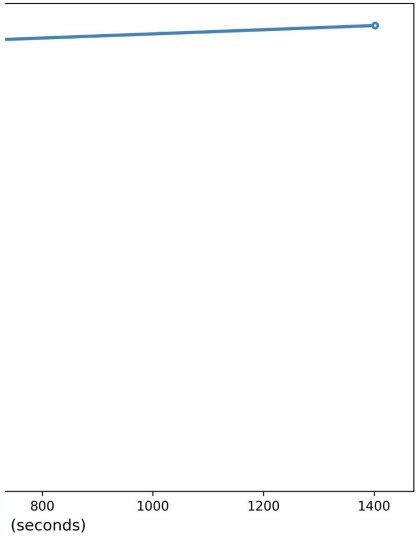
Diffing: 2021->25



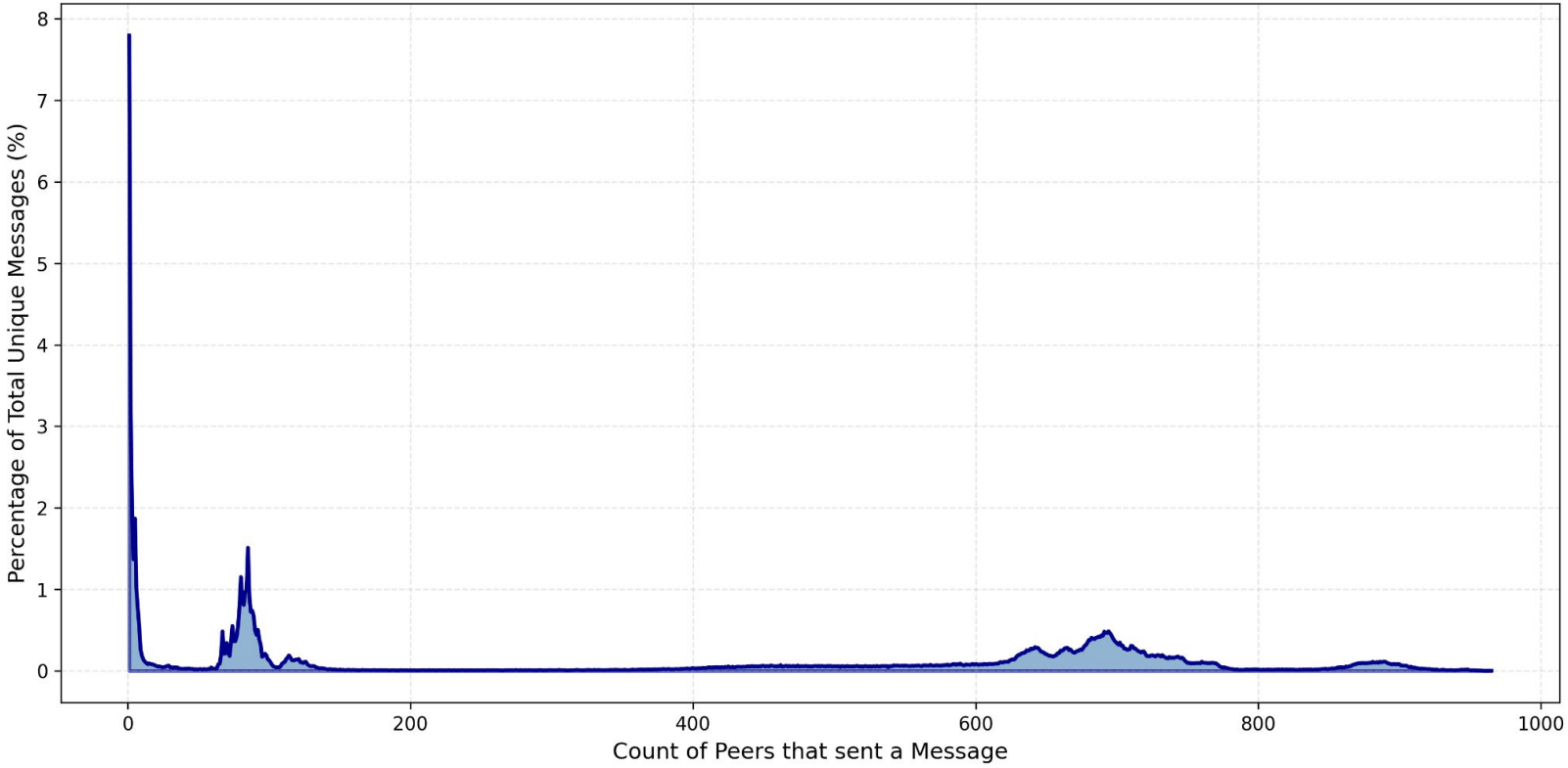


Diffing: 2021->25

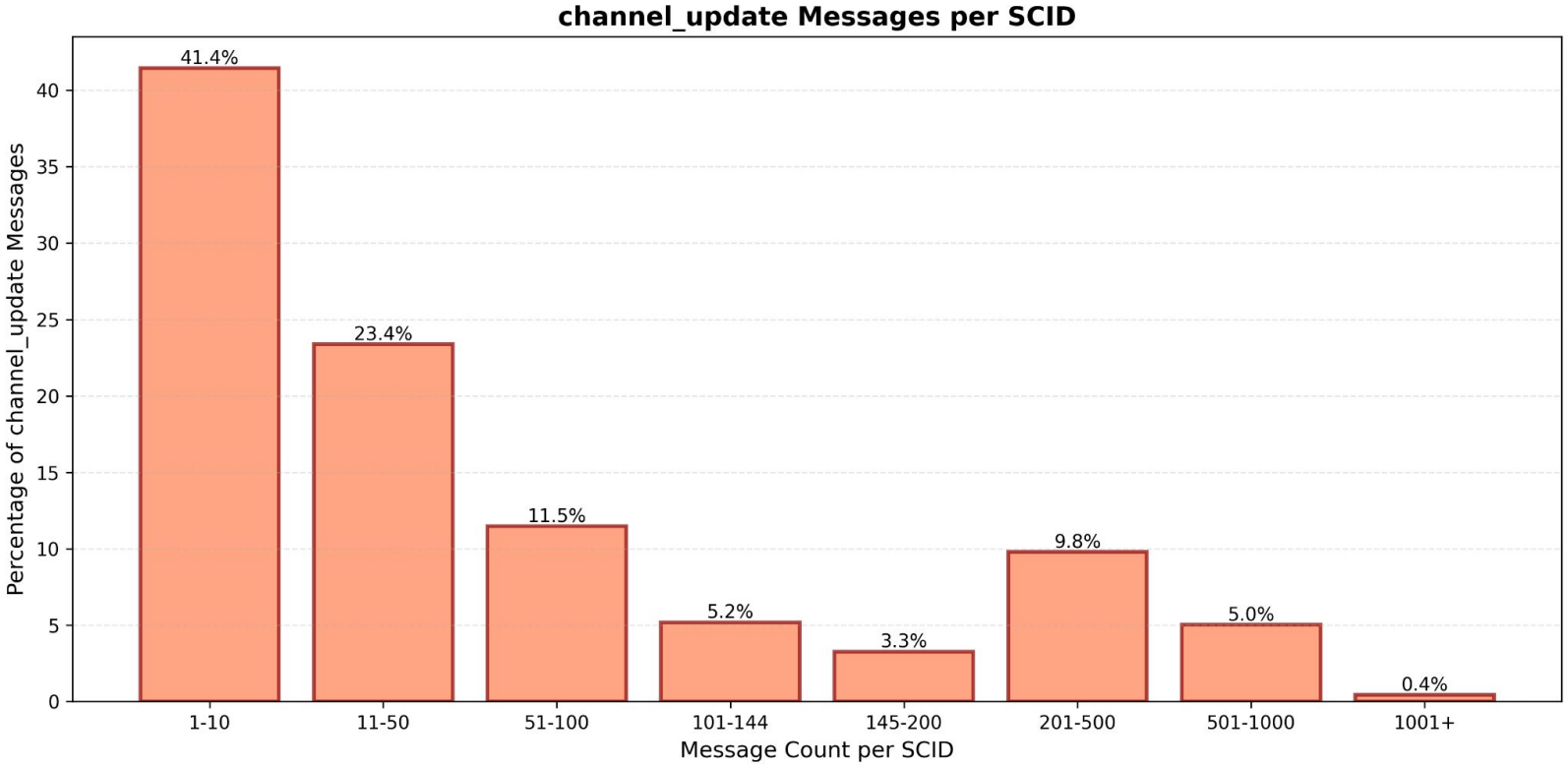




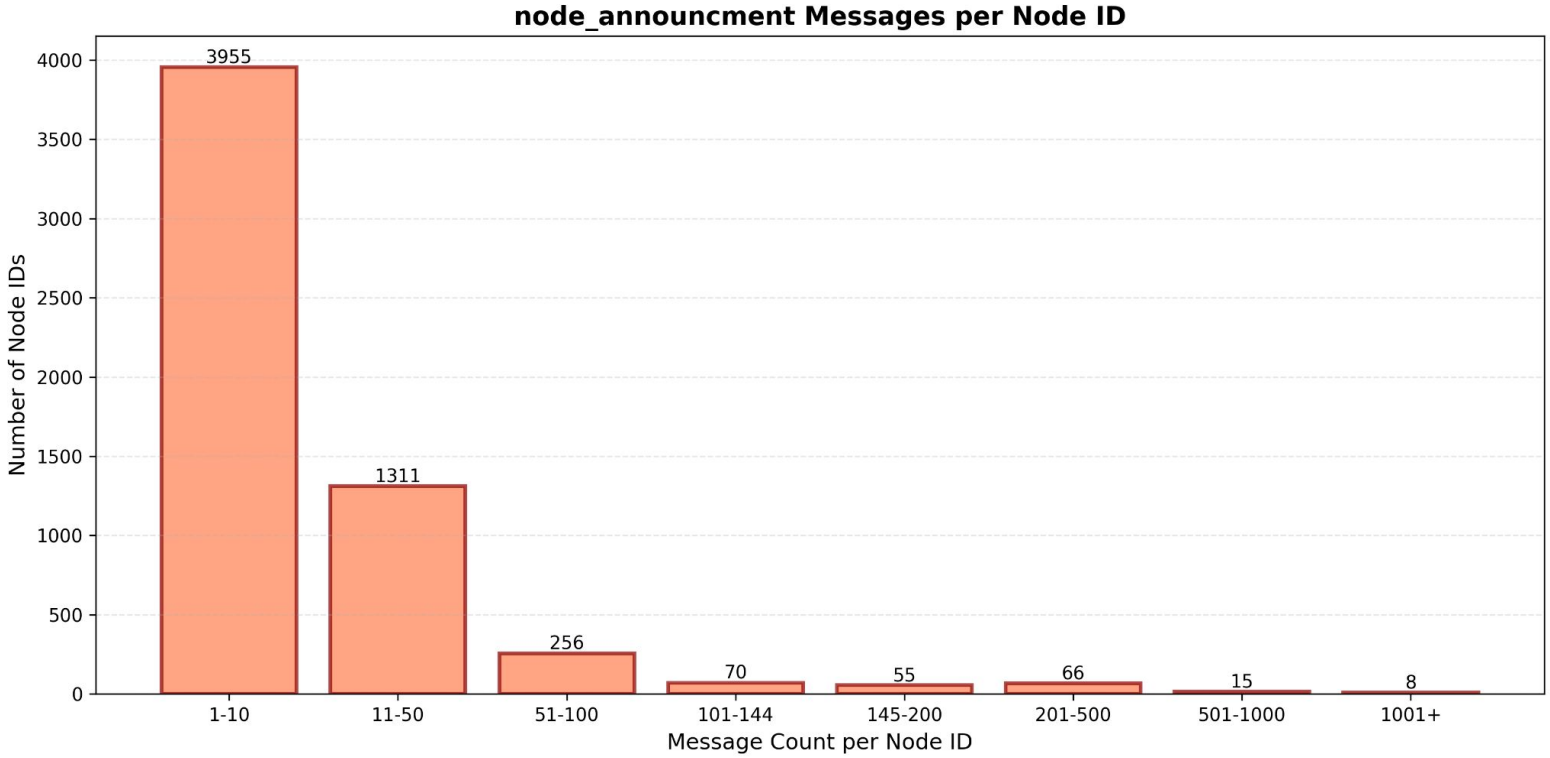
2025: Other patterns



2025: Other patterns



2025: Other patterns



2025: Who dis

Node: PiLN-Chiba≡ƒî | Magma Ready≡ƒïï

Follow

Public Node

Capacity

1.49434827 BTC (0.039%)
149,434,827 sat
$177,602.77

Channel Count

35 (0.081%)

Connected Node Count

35 (0.280%)

Color

#3399ff

IP Addresses

sshyttolzux2pw2osu772s7epc626gdv7qtejevwtzwmuyomz7upcmad.onion:9735
106.73.189.224:53441

Overview | Channels | Statistics | History | Neighborhood | Monitor

Public Key: 03910da61c1b42e135f134ed92a537c758d1edac5436efbec5ee8cec1928e1a095

03910da61c1b42e135f134ed92a537c758d1edac5436efbec5ee8cec1928e1a095@sshyttolzux2pw2osı

This node is a Japanese cryptocurrency geek node💎🙌 This node is automatically rebalancing its channels 24/7 by a script. Therefore, if you can get the channel connected with low fee, it may be routed by rebalancing. The power supply and network equipment are connected to a UPS, so there is no need to worry about going offline due to power failure. Please feel free to connect and expand your network!

Hardware

Raspberry Pi 4 Model B 4GB RAM

Networking

Fiber 1Gbps

2025: Other patterns

- Redundancy ratio: Average count of how many peers sent the same message
- channel_update: 447.8
- node_announcement: 407.5
- channel_announcement: 8.2 **(!)**

Minisketch: TL;DR

Bandwidth-Efficient Transaction Relay in Bitcoin

Gleb Naumenko
naumenko.gs@gmail.com
University of British Columbia

Gregory Maxwell
greg@xiph.org

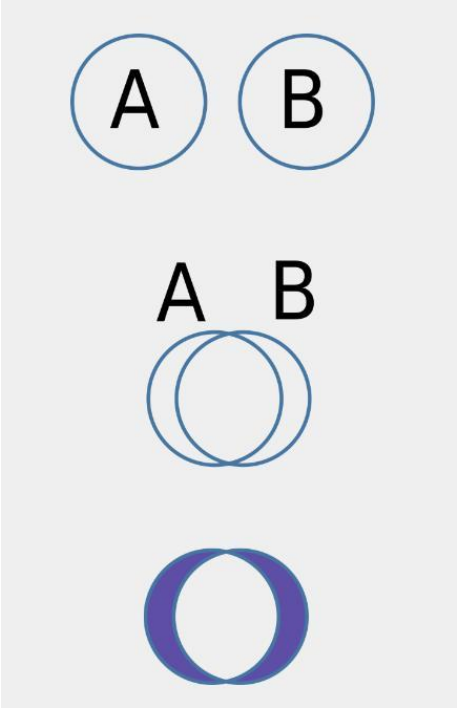
Pieter Wuille
pwuille@blockstream.com
Blockstream

Alexandra Fedorova
sasha@ece.ubc.ca
University of British Columbia

Ivan Beschastnikh
bestchai@cs.ubc.ca
University of British Columbia

Minisketch: TL;DR

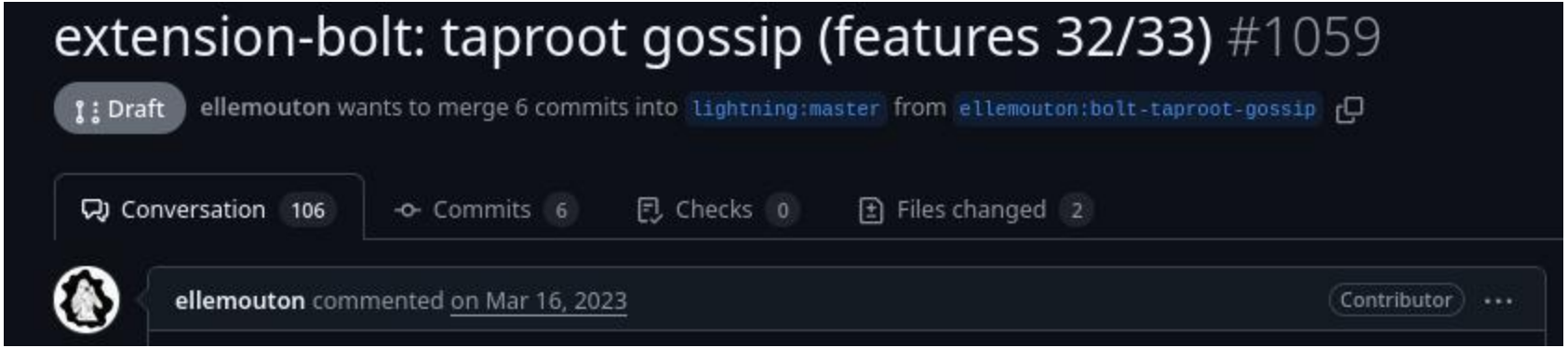
- Map elements to short (2-64 bit) collision-resistant IDs (SipHash or similar)
- Compute a sketch of some ‘capacity’
- Exchange sketches, reconcile
- Request differences, increase capacity, or fall back to something else



L1 TXs vs. LN gossip

- TXs are time-sensitive
- Source of TX broadcast is not public; should stay ‘private’
- Gossip messages include source information!

Gossip v2 (1.5?)


extension-bolt: taproot gossip (features 32/33) #1059
Draft
ellemouton wants to merge 6 commits into lightning:master from ellemouton:bolt-taproot-gossip
Conversation 106
Commits 6
Checks 0
Files changed 2
ellemouton commented on Mar 16, 2023
Contributor

Gossip v2 (1.5?)

- Support for existing P2WSH channels
- Add support for communicating about P2TR channels
- Rate limit messages with block height
- Use TLV format, with dedicated signed range **(!)**
- Leave room for optional, larger payloads like SPV proofs

Set Reconciliation Parameters

- Function to map messages to short IDs
- Sketch Capacity
- Failure Behavior

Set Reconciliation Parameters

- Function to map messages to short IDs
  - SipHash(signed TLV portion)
  - Multiple messages would have the same ID; differentiate by peer service bits?
  - Include the full TLV, and filter out fields per-peer?

- Avoid persistent set differences while propagating optional data

Set Reconciliation Parameters

- Sketch Capacity
    - Informed by traffic patterns; this can adjust at runtime / for each new reconciliation round
- Failure Behavior
    - Append another sketch, then fall back to direct query?

Future Work

- https://github.com/jharveyb/gossip_observer
- Continuous monitoring with multiple nodes, fewer connections
- New archive for collected data; what other fields are useful?
- Measure performance for ‘self’ messages like periodic announcement
- Ask me about your gossip!
- Finding Bad Neighborhoods ™