



Insights from block propagation in the Ethereum P2P network

DevCon 2024, Bangkok



Mikel Cortes

ProbeLab @ Shipyard

Which role does GossipSub play in Ethereum?

GossipSub is one of the **central pieces** of Ethereum:

- **Propagates** all validator **duties**
- Essential for **chain-finality**
- **Every node** runs it
- **Key component** in Ethereum's **scalability roadmap** (PeerDAS / FullDAS)



Why is it important to monitor it then?

Many external vectors affect the network performance:

- Block building **time-games**
- **Size of the block** content
- **Geographical location** of the nodes

GossipSub still provides:

- **Fast propagation** of blocks → reduce late block arrival
- **Resilience**

The ecosystem leans towards GossipSub with **more topics** and larger messages (PeerDAS)



What are the current limitations?

GossipSub **enforces some** soft/indirect **limitations**:

- Redundant information or **duplicate messages**
- Direct impact on **bandwidth requirements**

PLUS: these effects are **proportional to** the number of **bytes we** want to **propagate**



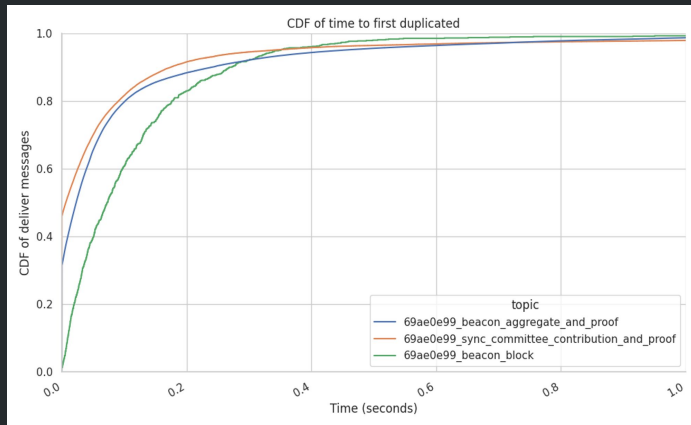
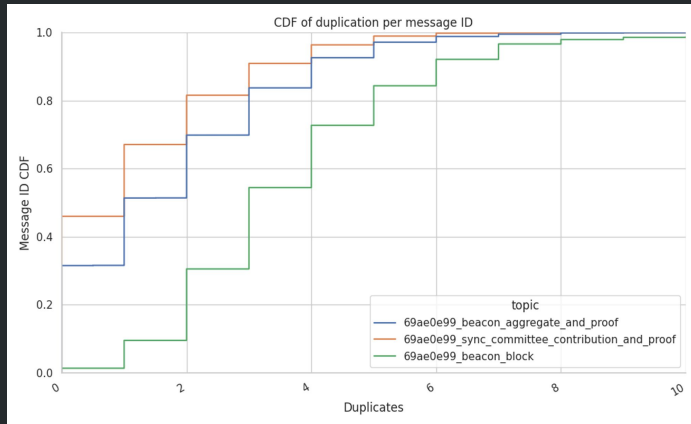
Message duplicates

Message **duplicates** are **guaranteed**

- **Bigger messages** mean **more duplicates**
- Bigger messages also generate **larger action times**

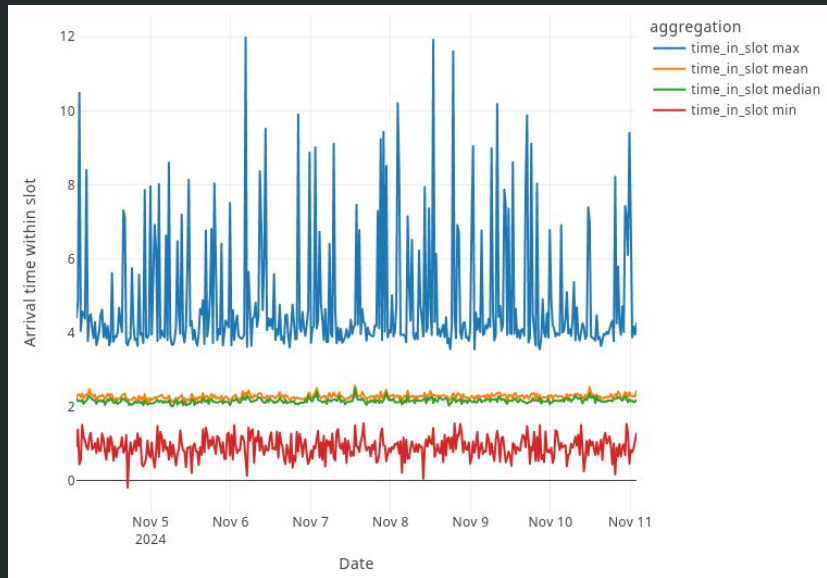
There are some optimizations:

- **IDONTWANT** control messages
- **Topic observation**

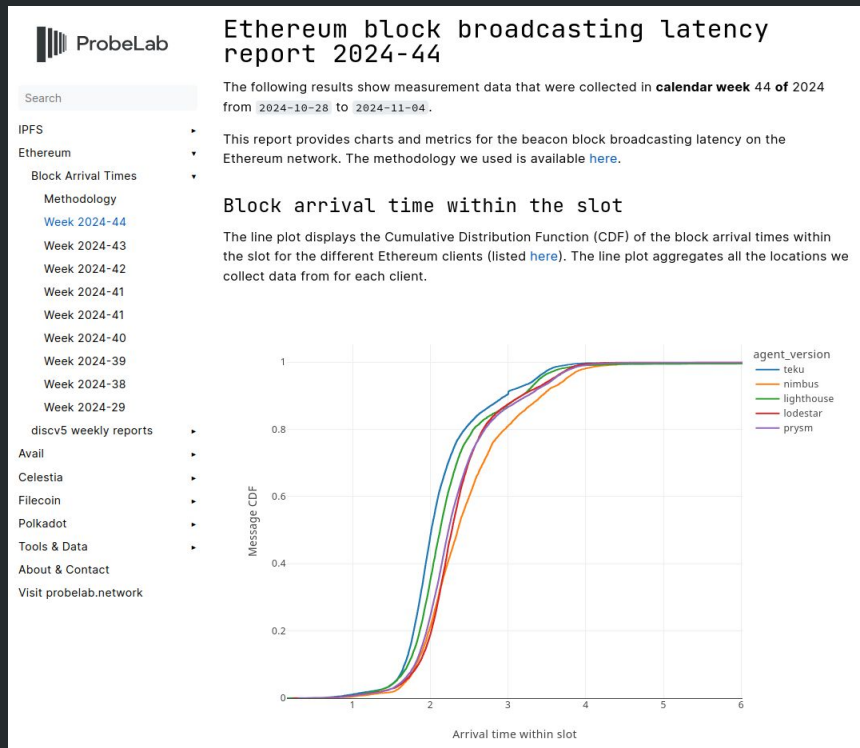


Impact of GossipSub to Block propagation

- **Monitoring** block arrival times is **crucial to spot** major network limitations
- **Constant monitoring** of the arrival times **allows identifying** the **performance benefits** of protocol upgrades



Impact of GossipSub to Block propagation



More info in our public reports

Mesh Stability
GRAFTs & PRUNES



GossipSub Effectiveness
IHAVES & IWANTS



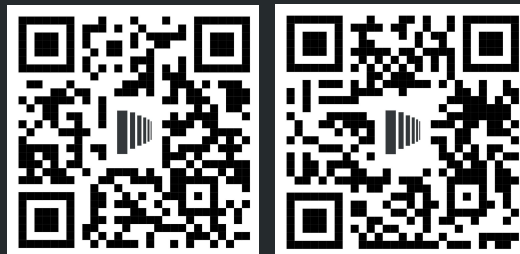
MessageMetrics
DUPLICATES



GossipSub
BANDWIDTH



CL Block
ARRIVAL TIMES





Q & A

Email: mikel@probelab.io

TG: [@cortze](#)