

Summary & TL;DR

The ProbeLab team (<https://probelab.io>) is carrying out a study on the performance of Gossipsub in Ethereum's P2P network. Following from our previous post on the "[Effectiveness of Gossipsub's gossip mechanism](#)", in this post we investigate the frequency of GRAFT

and PRUNE

messages and the dynamicity in terms of session duration and network stability that results from these protocol primitives. For the purposes of this study, we have built a tool called Hermes ([GitHub - probe-lab/hermes: A Gossipsub listener and tracer.](#)), which acts as a GossipSub listener and tracer. Hermes subscribes to all relevant pubsub topics and traces all protocol interactions. The results reported here are from a 3.5hr trace.

Study Description:

The purpose of this study is to determine the frequency between GRAFT

and PRUNE

messages as a distribution and standard deviation per topic. We measure the session duration between our peer and other peers, split per client and attempt to identify any anomalies (e.g., too short connections) and potential patterns.

TL;DR:

Overall, we conclude that Gossipsub is keeping a stable mesh as far as the MeshDegree

goes

, barely exceeding the DLow

and DHigh

thresholds of 6 and 12, respectively, despite increased dynamicity at times (i.e., increased numbers of GRAFTs

and PRUNEs

). Teku nodes always tear down connections to our node within a few seconds (or less) and fail to keep any connection running for longer periods of time (even minutes).

This fact is likely to lead to some instability, which, however, doesn't seem to be impacting the correct operation of the rest of the network.

For more details and results on Ethereum's network head over to <https://probelab.io> for discv5 Weekly Network Health Reports.

Background

GossipSub is the most widely used libp2p PubSub routing mechanism. Gossipsub includes enhanced PubSub routing that aims to reduce the bandwidth by maintaining fewer connections per subscribed topic (through the mesh) and sending some sporadic message metadata for resilience purposes (through the gossip mechanism). This ensures that despite the message being broadcasted using the shortest latency path (mesh), the peers still have the backup of sharing msg_ids

(gossip) to prevent missing messages that have already propagated through the network.

Because GossipSub reduces the overall number of connected peers by topic, known as meshes, having enough peers on these is crucial for efficient message broadcasting in any network. These meshes work as a connectivity sub-layer under the libp2p connections, as there is no direct map between 1_libp2p_peer_connection ↔ 1_mesh_peer_connection.

This differentiation provides efficiency to the protocol, as a peer doesn't need to "waste" bandwidth sending full messages to more peers than the needed ones while avoiding to spam peers who already shared the same message.

However, this means that the routing mechanism has to control how many peers it is connected to, or how many it needs to GRAFT

(add) or PRUNE

(remove) for each mesh, complicating things a little more.

This report provides insights into the GRAFT

and PRUNE

events (Network Dynamicity), and RPC calls (Session Duration) that our Hermes

-based node [\[link to Hermes GH repo\]](#) could track over 3.5 hours while participating in the Holesky testnet.

Results

Add and Remove peer events

The stability of the network's mesh relies heavily on the connections that the libp2p host keeps open. The following graph shows the number of ADD_PEER

and REMOVE_PEER

that the Hermes node tracked during the 3.5 hour run.

[

add-remove-peers-overall

1000×600 92.1 KB

](https://ethresear.ch/uploads/default/original/3X/6/9/69d25298153741ab11ad06836c18a6a5e46d85a2.png)

From the plot, we don't see any particularly inconsistent behaviour or pattern that stands out. The number of connections and disconnections at the libp2p host remains relatively stable around ~40 events per minute.

GRAFT

and PRUNE

events

GRAFT

and PRUNE

messages define when a peer is added or removed from a mesh we subscribe to. Thus, they directly show the dynamics of the peer connections within a topic mesh.

The following graph shows the number of GRAFTs

and PRUNEs

registered by the Hermes

node.

[

graft-prune-all-topics

1000×600 55.4 KB

](https://ethresear.ch/uploads/default/original/3X/b/7/b727e2a934e16138448ad37108aad42f831ed244.png)

We have also split this down by topic, and produced related plots, which however, we don't include here, as the split is roughly equal among topics.

We observe that the number of recorded events spikes after the ~2.5 hours of Hermes operation, jumping from peaks of ~100 events per minute to peaks of up to 700-800 events per minute.

Correlation with GRAFT

and PRUNE

RPC events

After 2 hours of the node being up online, we can see that the GossipSub host tracks mesh connectivity at a higher frequency than during the first hours. Since the RPC calls can include multiple topic GRAFTs

or PRUNEs

, these events correspond to the sum of original sent and received RPC calls.

Taking a closer look at the origin of those messages with the following plot, we can see that the most significant number of events belong to RECV_PRUNE

and SENT_GRAFTs

.

[

graft_prune_rpc_collection_all_graft_and_prune_rpc_calls

1000×600 55.1 KB

](<https://ethresear.ch/uploads/default/original/3X/5/0/50d7b77597f2824a7f9a9c6559bea2f78cfe1b08.png>)

As before, we have also split this down by topic, and produced related plots, which however, we don't include here, as the split is roughly equal among topics.

We can see that there is some perturbation to the stability of the meshes due to increased number of GRAFT

and PRUNE

events at the end of the trace period. There are two possible explanations for this:

- Remote nodes are dropping our connection for whatever reason (low peer score, or just sporadic PRUNE

messages because their mesh is full), and our Hermes

node counters this drop of peers by sending more GRAFT

messages to keep it up with the MeshDegree = 8

.

- We are sending too many GRAFT

messages to rotate our peers and test the connection to other nodes in the network, which is countered by the remote peers by sending PRUNE

messages as their mesh might be full already. Note that Hermes

has an upper limit of 500 simultaneous connections, so we expect it to have a higher range of connected peers than other nodes in the network.

In any case, we do not consider this to be an alarming event and the increased number of events might just be due to increased network activity. We will verify that this is not an alarming case through further experiments in the near future, where we'll collect traces for a longer period of time.

Mesh connection times

It is important to have a mix between steady connections and some rotation degree of these ones within each of the topic meshes. In fact, this level of rotation is the one that can guarantee that a node doesn't end up eclipsed by malicious actors (although GossipSub-v1.1 does have the gossiping feature as well to overcome eclipse attempts).

The following graphs show the average connection times per peer and per mesh.

[

connection_stability_cdf_per_topic

1000×600 48.4 KB

](<https://ethresear.ch/uploads/default/original/3X/6/6/66fb050fc61e3129fb919dd9610a801b39b7ae25.png>)

Measuring which is the average connection stability at the node level can showcase weird behaviours from the network. In this case, our Hermes

run did measure dispersed results were:

- 80% of the peers drop the connection after a few seconds of establishing it, although it has to be noted that the high percentage here owes to the spikes we've seen towards the end of the trace period in this particular dataset. We do not expect this to be the normal behaviour under "steady state".

[

connection_stability_cdf_per_topic-80

1000×600 54.1 KB

](https://ethresear.ch/uploads/default/original/3X/a/a/aa2b04f9e67c1fa8016566a43b29bf371df1ea0e.png)

- 10% of the peers remain connected for a total of ~4 minutes.

[

connection_stability_cdf_per_topic-10

1000×600 52.5 KB

](https://ethresear.ch/uploads/default/original/3X/8/a/8a01eb6b12aa60b64754879178b983fb20e68bfa.png)

- the remaining 10% of the connections remain between ~5 mins and 1.6 hours.

However, these plots do not give the full picture. As we saw with the number of GRAFT

and PRUNE

messages, there is a time relation on these distributions. What this means is that just because 80% of the connections happen within the first second, doesn't really mean that they were recently distributed over the trace period. To provide more clarity, we plot the connection duration (in seconds) split in 30min windows over the 3.5hr trace period.

[

connection-duration-30min-window-total

1000×600 14.1 KB

](https://ethresear.ch/uploads/default/original/3X/d/7/d7497c137c7f6a4f77e93f3c5bef245db964bb5.png)

Correlating the sudden spikes of GRAFT and PRUNE towards the end of the trace period, we find the following: the connections established over the first 2 hours and a half were indeed longer in duration.

We have created graphs breaking down the connections by topic, by agent, and by both topic and agent for 30min windows. We present only one representative plot here, as we didn't see any behaviour that stands out, other than the following:

- Lodestar clearly maintains connections for longer periods of time, followed by Nimbus
- Teku nodes consistently disconnect almost immediately.

[

connection_duration-30min_user_agents_total

1000×600 38.6 KB

](https://ethresear.ch/uploads/default/original/3X/6/7/679381131aa608c213d2e3de19ca87ecfd44454e.png)

Both of our observations are also evident from the following table, which includes the percentiles of the total connected time per client (in seconds).

Percentiles

p25

p50

p80

p90

p99

Grandine

6.53
17.22
31.52
43.67
117.36
Lighthouse
9.94
19.82
36.25
74.89
570.02
Lodestar
2.06
7.12
1768.40
5855.87
7165.20
Nimbus
200.44
523.96
599.88
599.97
4157.40
Prysm
5.00
5.00
175.65
594.67
4322.25
Teku
0.10
0.13
0.64
1.72
5.85
Unknown
0.07

0.14

5.00

5.22

5.90

Resulting number of peers per mesh

It is important to highlight that despite the spike in network connectivity and the relevant RPC interactions our node has kept a stable range of connections within each of the topic meshes.

The following graph shows the total number of mesh connections for each topic, binned at 5 minute intervals.

[

graft_and_prune_events_collection_mesh_connectivity_for_topics

1000x600 106 KB

](<https://ethresear.ch/uploads/default/original/3X/d/c/dcccaabc9e3ab2c55c355ca18311035e560e9988.png>)

The range of connections per topic barely goes below 6, even at the last hour of data, where we've seen the spikes in GRAFT

and PRUNE

events. We can observe that:

- GossipSub is doing a good job at keeping the range of connections around the `gossipSubD=8`

(between the `gossipSubDlo=6`

and the `gossipSubDhi=12`

).

- This ratio of connections per topic does see a small drop on the last hour of the run, matching the sudden spike of `RECV_PRUNES`

seen in earlier plots.

- The drop in the number of connections during the last hour does seem related to the fact that the connections opened during this period were actually way shorter than the previous ones.

Conclusions and takeaways

- Overall, we conclude that Gossipsub is keeping a stable mesh as far as the `MeshDegree`

goes

, barely exceeding the `DLow`

and `DHigh`

thresholds of 6 and 12, respectively, despite increased dynamicity at times (i.e., increased numbers of GRAFTs

and PRUNES

).

- Teku nodes always tear down connections to our node within a few seconds (or less) and fail to keep any connection running for longer periods of time (even minutes).

This fact is likely to lead to some instability, which, however, doesn't seem to be impacting the correct operation of the rest of the network (at least from what we can see so far).

- Our data shows a sudden spike in GRAFT

and PRUNE

events during the last hour of the study, despite the ADD_PEER

and REMOVE_PEER

events staying steady over the entire run.

This could suggest that Hermes

or the network is struggling to maintain mesh connections with other participants. * The spike in GRAFTs

and PRUNEs

generally comes from incoming PRUNE

events followed by subsequent outgoing GRAFT

events. It is still not clear what triggers this behaviour, i.e., if it's just an anomaly coming from the strict connectivity limits from remote peers that refuse connections quite often, or if it's Hermes

the one that spams remote peers with GRAFTs

more than it should and remote peers in turn respond with PRUNE

events.

- We conclude that this incident is interesting to keep an eye on (e.g., nodes end up with stable connections at the Host level, but struggling to keep a healthy number of connections in each mesh) and verify through a longer experiment. However, we do not consider this to be a critical incident, i.e., doesn't cause any other metric to suffer.
- The spike in GRAFTs

and PRUNEs

generally comes from incoming PRUNE

events followed by subsequent outgoing GRAFT

events. It is still not clear what triggers this behaviour, i.e., if it's just an anomaly coming from the strict connectivity limits from remote peers that refuse connections quite often, or if it's Hermes

the one that spams remote peers with GRAFTs

more than it should and remote peers in turn respond with PRUNE

events.

- We conclude that this incident is interesting to keep an eye on (e.g., nodes end up with stable connections at the Host level, but struggling to keep a healthy number of connections in each mesh) and verify through a longer experiment. However, we do not consider this to be a critical incident, i.e., doesn't cause any other metric to suffer.