Gossip Service in a Solana Validator

The Gossip Service acts as a gateway to nodes in thecontrol plane . Validators use the service to ensure information is available to all other nodes in a cluster. The service broadcasts information using agossip protocol .

Gossip Overview

Nodes continuously share signed data objects among themselves in order to manage a cluster. For example, they share their contact information, ledger height, and votes.

Every tenth of a second, each node sends a "push" message and/or a "pull" message. Push and pull messages may elicit responses, and push messages may be forwarded on to others in the cluster.

Gossip runs on a well-known UDP/IP port or a port in a well-known range. Once a cluster is bootstrapped, nodes advertise to each other where to find their gossip endpoint (a socket address).

Gossip Records

Records shared over gossip are arbitrary, but signed and versioned (with a timestamp) as needed to make sense to the node receiving them. If a node receives two records from the same source, it updates its own copy with the record with the most recent timestamp.

Gossip Service Interface

Push Message

A node sends a push message to tell the cluster it has information to share. Nodes send push messages toPUSH_FANOUT push peers.

Upon receiving a push message, a node examines the message for:

1. Duplication: if the message has been seen before, the node drops the message
2. and may respond withPushMessagePrune
3. if forwarded from a low staked node
4. New data: if the message is new to the node
5.
    - Stores the new information with an updated version in its cluster info and
6.
    - purges any previous older value
7.
    - Stores the message inpushed_once
8.
    - (used for detecting duplicates, purged
9.
    - afterPUSH_MSG_TIMEOUT * 5
10.
    - ms)
11.
    - Retransmits the messages to its own push peers
12. Expiration: nodes drop push messages that are older thanPUSH_MSG_TIMEOUT

Push Peers, Prune Message

A node selects its push peers at random from the active set of known peers. The node keeps this selection for a relatively long time. When a prune message is received, the node drops the push peer that sent the prune. Prune is an indication that there is another, higher stake weighted path to that node than direct push.

The set of push peers is kept fresh by rotating a new node into the set everyPUSH_MSG_TIMEOUT/2 milliseconds.

Pull Message

A node sends a pull message to ask the cluster if there is any new information. A pull message is sent to a single peer at random and comprises a Bloom filter that represents things it already has. A node receiving a pull message iterates over its values and constructs a pull response of things that miss the filter and would fit in a message.

A node constructs the pull Bloom filter by iterating over current values and recently purged values.

A node handles items in a pull response the same way it handles new data in a push message.

Purging

Nodes retain prior versions of values (those updated by a pull or push) and expired values (those older thanGOSSIP_PULL_CRDS_TIMEOUT_MS) inpurged_values (things I recently had). Nodes purgepurged_values that are older than5 * GOSSIP_PULL_CRDS_TIMEOUT_MS .

Eclipse Attacks

An eclipse attack is an attempt to take over the set of node connections with adversarial endpoints.

This is relevant to our implementation in the following ways.

- Pull messages select a random node from the network. An eclipse attack onpull
- would require an attacker to influence the random selection in such a
- way that only adversarial nodes are selected for pull.
- Push messages maintain an active set of nodes and select a random fanout for
- every push message. An eclipse attack onpush
- would influence the active set
- selection, or the random fanout selection.

Time and Stake based weights

Weights are calculated based ontime since last picked and thenatural log of thestake weight .

Taking theln of the stake weight allows giving all nodes a fairer chance of network coverage in a reasonable amount of time. It helps normalize the large possiblestake weight differences between nodes. This way a node with lowstake weight , compared to a node with largestake weight will only have to wait a few multiples of ln(stake) seconds before it gets picked.

There is no way for an adversary to influence these parameters.

Pull Message

A node is selected as a pull target based on the weights described above.

Push Message

A prune message can only remove an adversary from a potential connection.

Just likepull message , nodes are selected into the active set based on weights.

Notable differences from PlumTree

The active push protocol described here is based onPlum Tree . The main differences are:

- Push messages have a wallclock that is signed by the originator. Once the
- wallclock expires the message is dropped. A hop limit is difficult to
- implement in an adversarial setting.
- Lazy Push is not implemented because it's not obvious how to prevent an
- adversary from forging the message fingerprint. A naive approach would allow
- an adversary to be prioritized for pull based on their input.Previous Validator's Transaction Validation Unit (TVU) Next Validator Runtime