Overview

- DRiP is a HTTP based protocol for sharing registry type of information between interconnected nodes across a network.
- It uses a gossip protocol for complete distribution across interconnected nodes.
- It incorporates a voting mechanism to avoid conflicting data updates or race conditions.
Distributed Mesh
Transactions

- Two basic transactions
  - Update - A node has new or modified key-value data and would like to update peer nodes
  - Sync - A node is either newly established or was in an inactive state for a period of time and requests a peer to provide a full update of data to make sure it is fully synchronized with network.
Voting and Commit Phases for Update

• When initiator node has new data, it initiates an Update

• Update consists of a two-phase commit procedure to avoid race conditions or potential error conditions

• Two phases are called:
  • voting phase
  • commit phase
Voting and Commit Phases for Update

1. Waiting For Events
2. Received Update From Peer Node
   - If key matches an in-progress update vote "no".
   - Otherwise, vote "yes".
3. Waiting For Response From Peer Nodes
4. Timer Expired
5. Received Votes From All Peer Nodes
6. Validating Votes
   - (If all Votes are "YES", propagate commit)
Authentication/Entitlement

• Took the approach that scope of this spec only has protocol for exchanging data

• Assumes any authentication or entitlement of write/read capability or permissions sits a layer above this protocol and/or in the key-value data model
Where do we go from here?

- No document for IETF95, but still committed to updating going forward and providing an open implementation (testbed effort planned later this year in ATIS as well)

- Propose a separate draft for the Auth/Security framework for managing and protecting the DRiP network, API credentials/tokens, revoking access, etc.

- Number/identity porting security could be another potential draft as well.

- I believe these things will evolve dependent on data model we agree on as well.