TCoin: A Crypto Currency based on Proof of Stake And Distributed Hash Table Networking

Naman Arora, Gurupad Hegde and Sourabh Gopal Parvartikar

Introduction

1. Tackle the Blockchain Trilemma of scalability, security and decentralization.
2. Distributed Hash Table (DHT) Networking:
    1. Scalability + Decentralization
    2. Fault Tolerance
3. Proof of Stake (PoS) concensus algorithm
    1. Security + Scalability
    2. More feasible than PoW
    3. Provides testbed for future targetted research
4. Shallow transaction history, thus higher transaction rate

Milestones

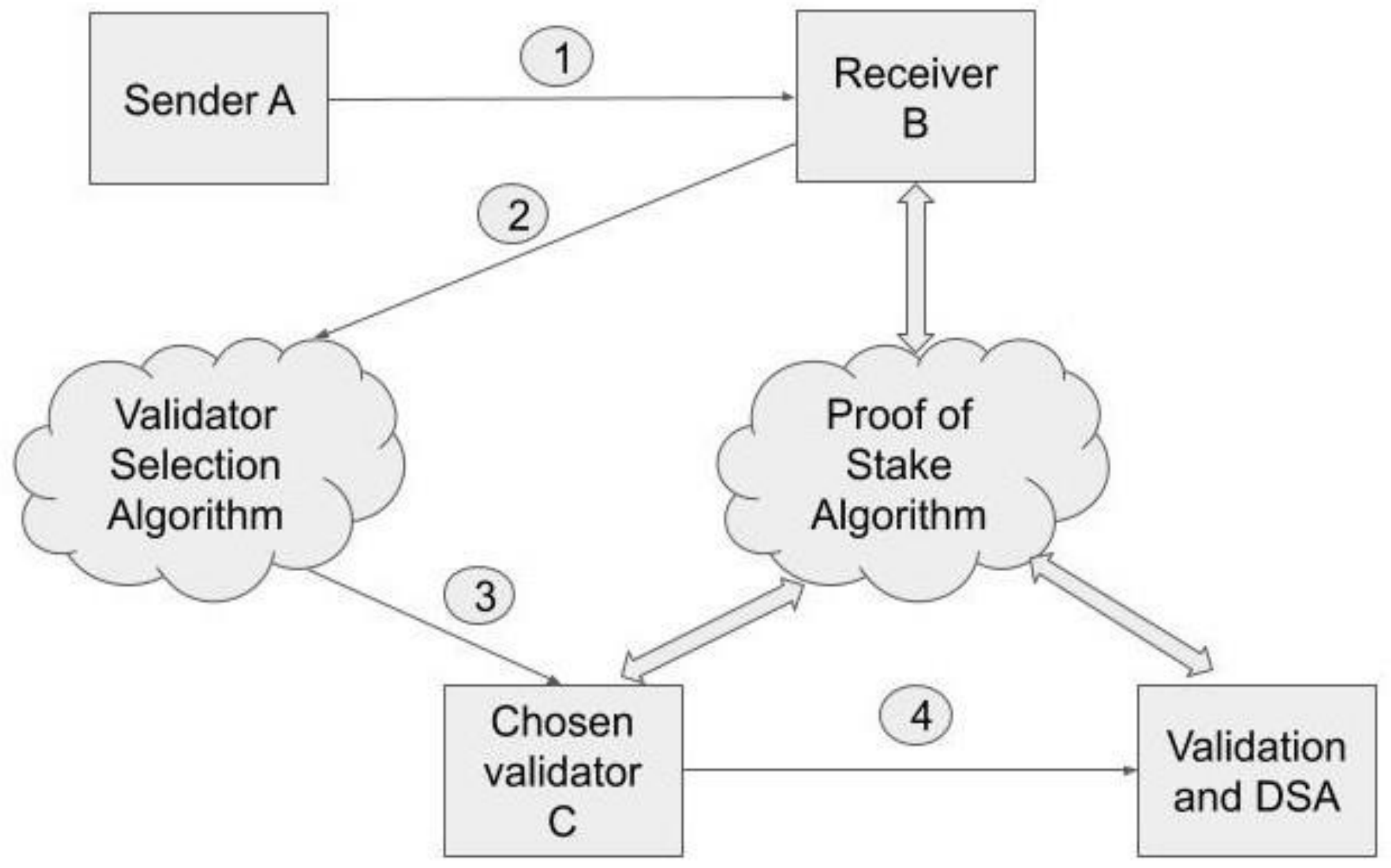
1. Pre-midterm (Accomplished)
    1. Chioce of programming enviornment
    2. Brainstorming the nuances of the architecture
    3. Completion and testing of DHT network as an API
2. Post-midterm (Underway)
    1. Completion and testing of PoS
    2. Itegration and testing of PoS and DHT using a barebones wallet like node entity

Outcomes till Mid-term

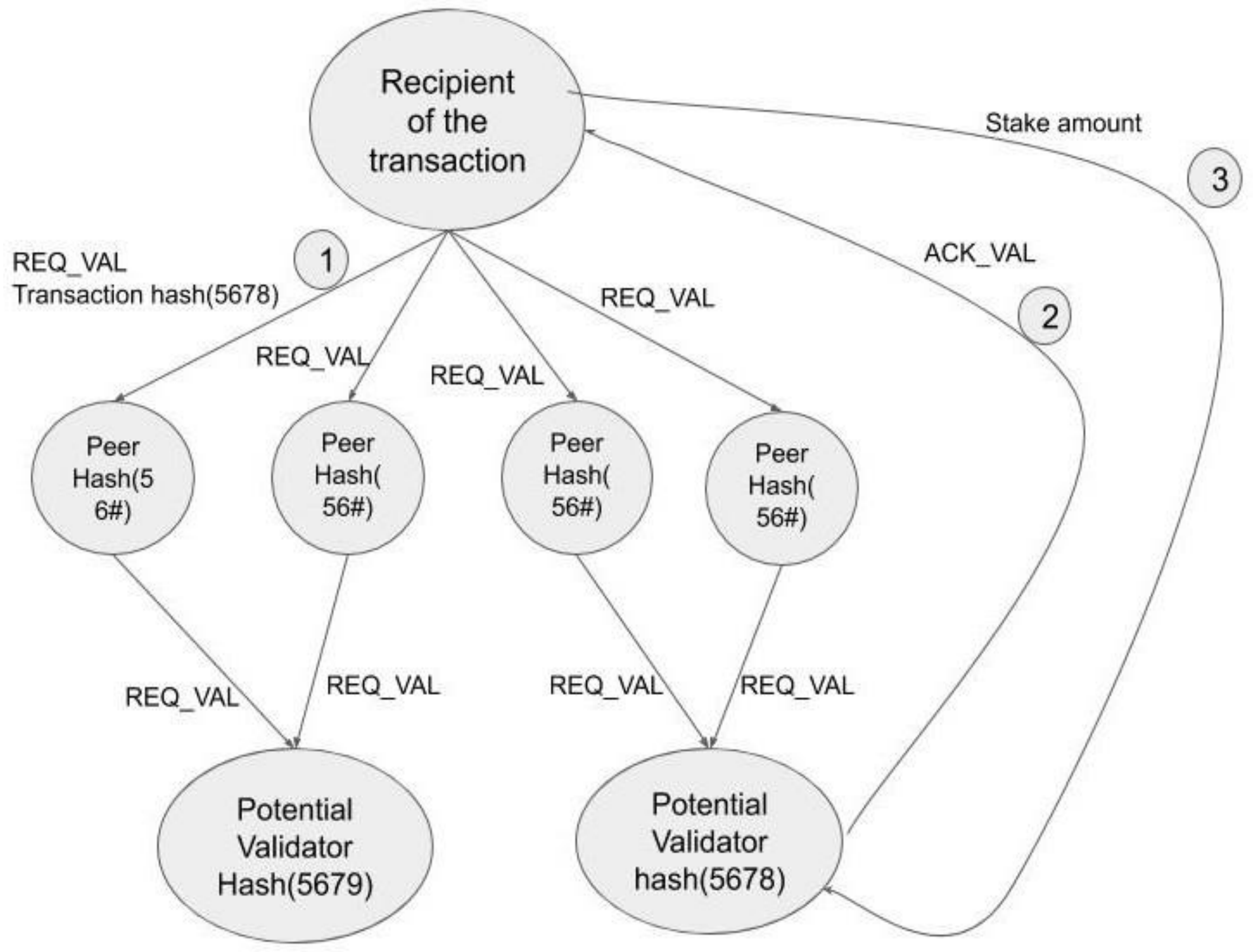
1. Elixir is programming language of choice
2. Testing of DHT network shows fault tolerant behavior
3. Work can be tracked on
    - GitLab Mirror
    - GitHub Mirror

Methodology

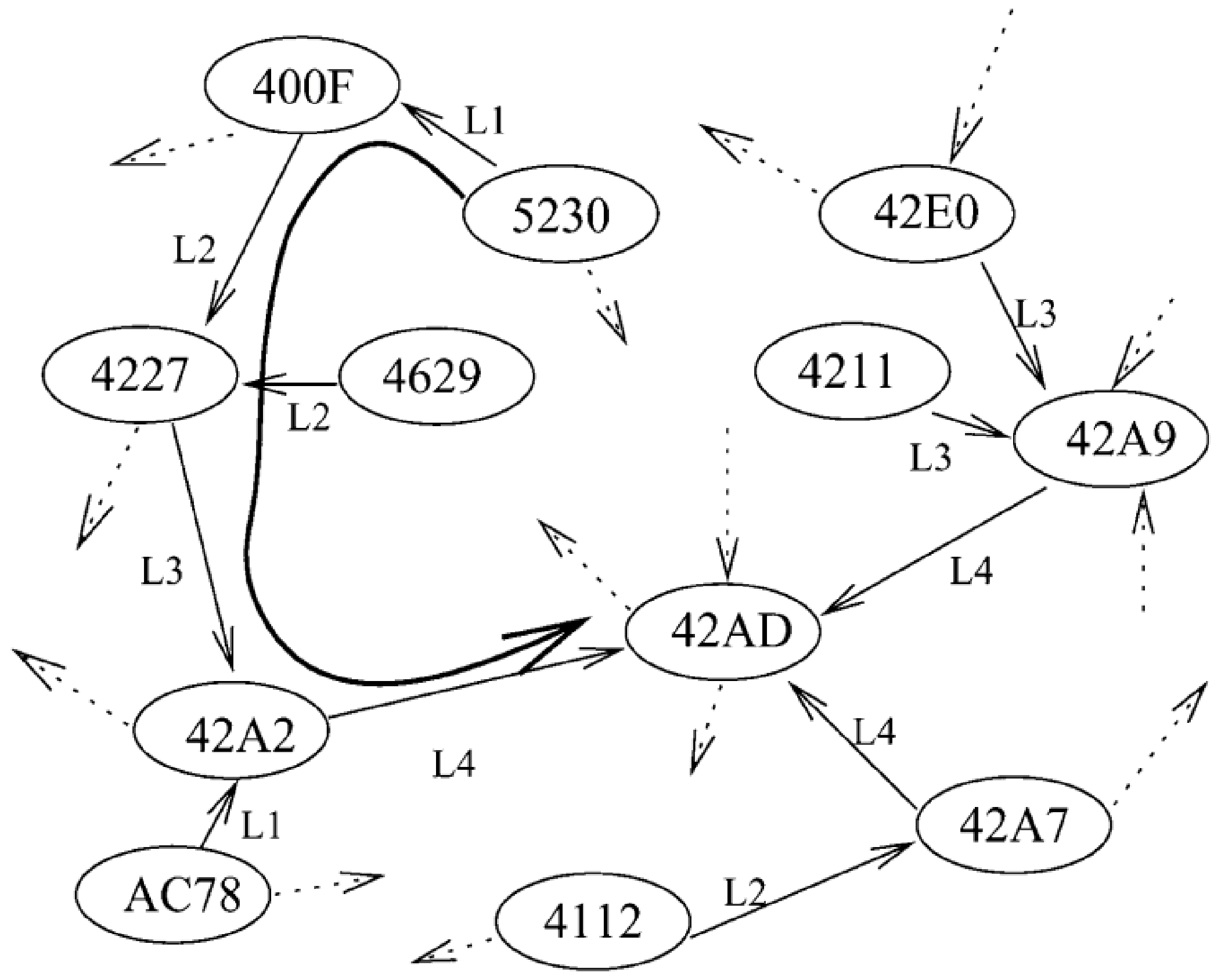
1. Transaction Algorithm:
    1. Validator is oblivious to transaction amount, thus cannot form bias.
    2. 50% of transaction amount is put as stake with 5% of it as incentive.
    3. Transactions and Node addresses have associated unique hashes, cryptographically secure and from a same pool.
    4. Validator receives prior transaction hashes and tracks them down using network layer for validation purpose.
    5. After each successful transaction, receiver publishes the new transaction as its own and validator purges the payee's prior transactions.
    6. Validator's resources are wasted if transaction fails, but stake is released as is.
    7. Stake is confesticated by receiver if validator is uuntrustworthy, compounded as incentive for next validator. (determining the trustworthyness of validator is still in progress)
2. Validator selection algorithm:
    1. The receiver publishes a request for validator using network API
    2. The first node to respond and able show enough funds for stake is selected by receiver
    3. Transaction amount is undisclosed to validator until locked
    4. Receiver acts as a validator for the micro-transaction of stake confestication after validator selection
    5. This algorithm can in involked in the event of:
        1. A validator is required for for a transaction
        2. A node needs to renew funds before network reboot (see below)
3. Transaction Renewal Process:
    1. After a certain time period, each transaction is purged.
    2. Each node is responsible of renewing its transactions before network purge.
    3. Purged transactions are irrecoverable.
    4. Validator gets its stakes renewed by the receiver holding them if network happens to reboot during the process.
4. Network description (Network API):
    1. Root Node of an entity is a node whose hash is closest to that entity's hash
    2. Pointer to an object is a data structure that stores information where an object with particular hash can be found
    3. Publish requires the publisher to distribute the object's pointers to all the nodes in the way of the object's root node.
    4. Unpublish removes all the pointers of that object
    5. Route to Object requires a node to reach any pointer of the object while routing to its root node



Transaction Algorithm



Validator Selection Algorithm



DHT Routing example