Phase 2: DiemBFT v4 Consensus Algorithm

Building and Running Instructions

First, ensure that DistAlgo and PyNaCl are installed:

```
pip install pyDistAlgo pynacl
```

You can run all the test cases with this command:

```
./run_tests.sh
```

This command runs a particular test by name (faulty_dropouts, in this case):

```
python -m da --message-buffer-size 80000 --no-log src/run.da
faulty_dropouts
```

Configuration File Format

Configuration files are DistAlgo source files that contain two global variables: “config” and “failure_config.”

Config is a named tuple with the following fields:
num_replicas: int
num_faulty: int
workloads: list of Workloads
transmission_delay_bound: float
seed: float

Workloads specify how the clients send requests and await responses. If any of the workloads in a Config time out, the test fails. Workload is a named tuple with the following fields:
type: WorkloadType
count: int (the number of transactions)
num_clients: int (used for multiple clients with the same workload)
timeout: float
delays: list of floats

WorkloadType:
retransmit - Clients wait for f + 1 acknowledgements of each message before sending the next one. Exponential backoff is used to determine an appropriate delay between retransmissions
flood - Clients send all of their requests at once, and they do not retransmit them
timed - Clients send requests at timed intervals according to the “delays” field of the Workload. Messages are not retransmitted.

FailureConfig:
failures: list of Failures
seed: float

A Failure can be a message loss, message delay, or SetAttribute (Byzantine failure). A Failure is a named tuple with the following fields:
src: replicaID, “leader,” or wildcard
dest: replicaID, “leader,” or wildcard
msg_type: Proposal, TimeOut, Vote, or Wildcard
round: list of rounds in which the failure should occur
prob: float
val
attr

All of these types are defined in src/config.da

Phase 4: Implementation of Twins and Testing of diemBFT

To generate scenarios with different test generator configurations (all_random_configs, in this case):

```
python3.7 testgenerator.py generator_configs/all_random_config.py
```

To execute scenarios with different test executor configurations (conflicting_votes_bug, in this case):

```
python3.7 -m da --message-buffer-size 80000 src/testexec.da
executor_configs/conflicting_votes_bug.py
```