Meeting Notes 14-12-2021

Progress Update
Test harness update, transaction submission issues.

- Propose sequence maximum? Hard to find maximum, will pick a number (seq=2) as maximum and not delay any propose messages sent after.
- Thesis outline: Execution and schedules separate item in background. Encoding before operators. Combine background and related work.

Schedule comparison, metric is fine

Master registration system
First stage review date. Skip this date, fill in the rest.

Traces and executions
Event dependency. Two concurrent messages are dependent if their recipient is the same node, and independent otherwise. Check concurrency of message by tracking causal dependencies.

- Take all messages pertaining to consensus protocol. Even if we delay a subset of consensus messages.
- Causal dependencies:
    - Option 1: Keeping track of vector clocks. Increment vector clock of the node whenever it receives a message. Most likely requires a ripple code alteration, undesired.
    - Option2: Keep track of dependency in the scheduler module. Computationally intensive alternative: Dynamically determine the events that are concurrent and causal. Use time window for this.
      Space intensive alternative: Create a directed acyclic graph where each node is an event and edges represent (transitive) causal dependencies of an event to another.

      Might be too coarse. Look at ripple code for determining more fine-grained dependence of particular message types.

Once we have this dependency information, we can use this in the mutation operator to guide mutation to genes that are likely to change the trace.

Next steps

1. Implement dependency tracking to create traces.
2. Apply Levenshtein sequence ratio to said traces and determine quality of problem encoding
3. Compare fitness functions and pick
4. Use traces in mutation operator