**Synod Algorithm (without timeouts and retries)**

Proposer State:

- `maxPropNum`: largest proposal number seen so far

Acceptor State:

- `maxPrepare`: largest Proposal number for which it has responded to a proposal, initially 0
- `(accNum, accVal)`: largest numbered proposal that is has accepted, initially (null, null)

**Phase 1:**

- **Proposer:** picks new proposal number \( n \) that is larger than `maxPropNum` and updates `maxPropNum` sends `prepare(n)` to acceptors
- **Acceptor:** on receiving `prepare(n)`
  - if \( n > maxPrepare \)
    - respond to promise with `promise(accNum, accVal)`

**Phase 2:**

- **Proposer:** if receive `promise` from majority of Acceptors
  - \( v = accVal \) with largest `accNum` received on promise messages
  - if all `accVal = null`
    - proposer uses own value for \( v \)
  - sends `accept(n,v)` to Acceptors
- **Acceptor:** on receiving `accept(n,v)`
  - if \( n \geq maxPrepare \)
    - \((accNum, accVal) = (n,v)\)

**Learning a chosen value:**

- **Option 1:**
  - When acceptor accepts proposal, it sends `accepted(accNum, accVal)` to all Learners
  - Learner commits value when it receives same `accepted(accNum, accVal)` from a majority of Acceptors

- **Option 2:**
  - When Acceptor accepts a proposal, it sends `accepted(accNum, accVal)` to a Distinguished Learner
  - When Distinguished Learner same receives `accepted(accNum, accVal)` from a majority of Acceptors, it sends `commit(accNum,accVal) to all Learners`
  - When Learner receives `commit(accNum,accVal)` it commits the value