Synod Algorithm (Lamport)

Execution of Synod Algorithm (1)

1. Choose new proposal number \( m \).
   Send \( \text{prepare}(m) \) to all acceptors.

2. If \( m > \text{maxPrepare} \)
   \( \text{maxPrepare} = m \),
   reply with \((\text{accNum}, \text{accVal})\).

3. If receive response from majority
   choose value \( v \),
   send \( \text{accept} \) to all acceptors.
   Else, start over.

   \( v = \text{value with largest \( \text{accNum} \) number.} \)
   Only if all \( \text{accVal} \) values are \( \text{null} \), choose own value.

Execution of Synod Algorithm (2)

5. Send \( \text{ack} \) after saving \( \text{accNum} \) and \( \text{accVal} \)

6. If receive \( \text{ack} \) from majority
   send \( \text{commit}(v) \)

4. Record \( v \) in log.
**Byzantine Agreement Algorithm with Oral Messages** (Lamport, Shostak, Pease: 1982)

**Base Case: OM(0) \ (m = 0)**

1. Commander sends value v to every lieutenant.
2. Each lieutenant j returns v or RETREAT if no value received.

**OM(m) \ (m > 0)**

1. Commander sends value v to every lieutenant.
2. For each lieutenant j
   \[ v_j = \text{value Lieutenant j receives from Commander or RETREAT if no value received.} \]
   Lieutenant j acts as Commander to execute OM(m-1) \ (*to send v_j to n-2 other lieutenants*).
3. For all j, for all k \( \neq j \)
   \[ v_k = \text{value Lieutenant j received from Lieutenant k for step 2 (using OM(m-1)) or RETREAT if no value received.} \]
   Return majority\((v_1,v_2,...,v_{n-1})\).