

# Consensus in Synchronous System

Possible to achieve!

- For a system with at most  $f$  processes crashing
  - All processes are synchronized and operate in “rounds” of time
  - the algorithm proceeds in  $f+1$  rounds (with timeout), using reliable communication to all members. Round length  $\gg$  max transmission delay.
  - $Values^r_i$ : the set of proposed values known to  $p_i$  at the beginning of round  $r$ .
- Initially  $Values^0_i = \{\}$  ;  $Values^1_i = \{v_i\}$ 
  - for round = 1 to  $f+1$  do
    - multicast** ( $Values^r_i - Values^{r-1}_i$ ) // iterate through processes, send each a message
    - $Values^{r+1}_i \leftarrow Values^r_i$
    - for each  $V_j$  received
      - $Values^{r+1}_i = Values^{r+1}_i \cup V_j$
    - end
  - end
  - $d = \text{minimum}(Values^{f+1})$  // consistent minimum based on say id (not minimum value)