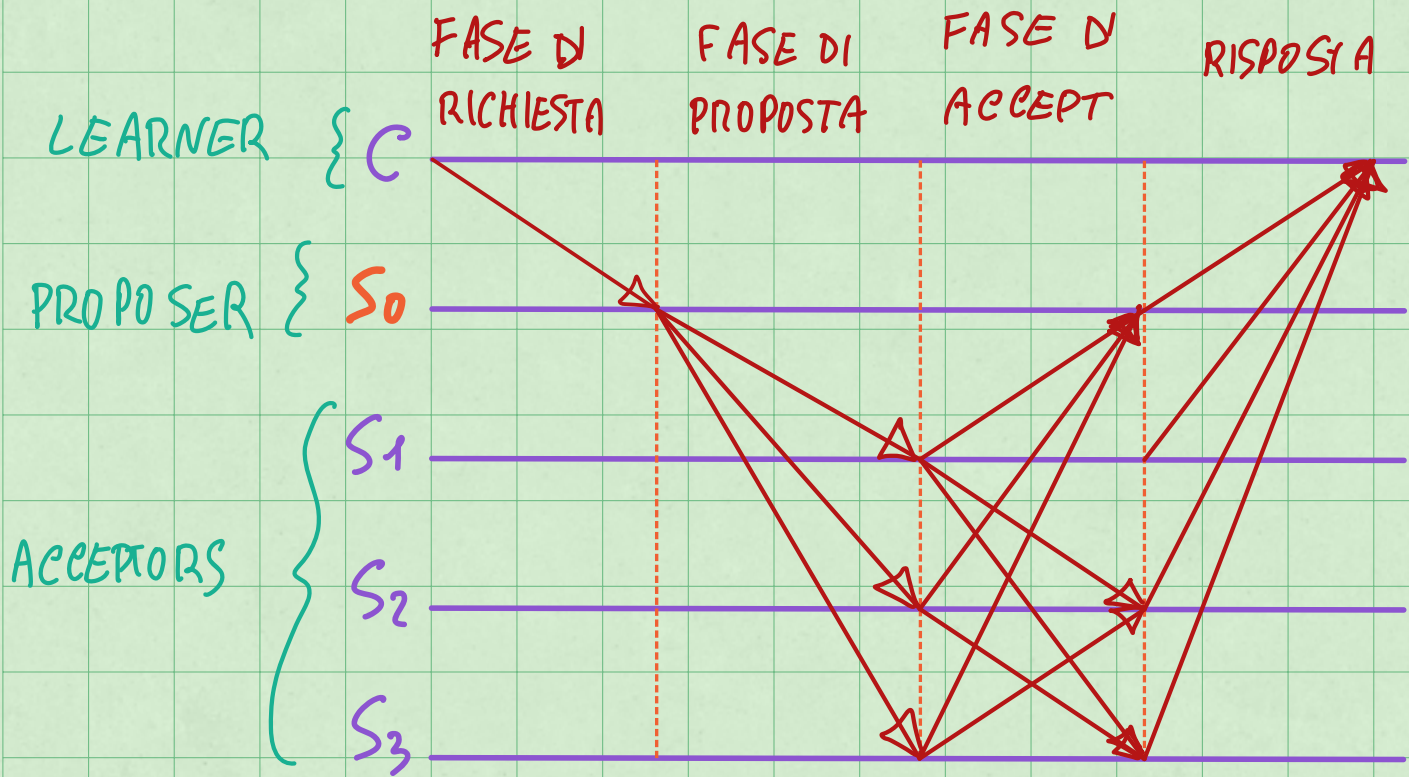


# PAXOS (SMR)

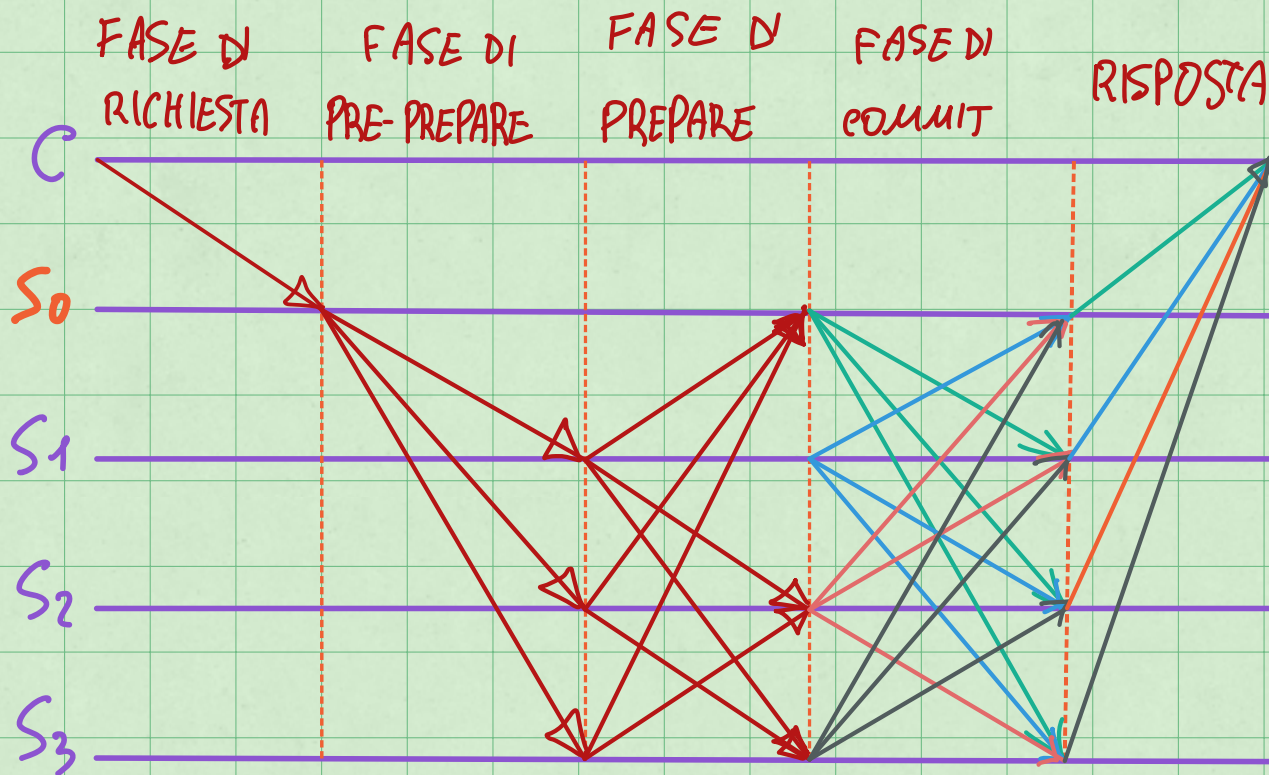


1. Il client invia un valore al server S<sub>0</sub> il quale ricopre il ruolo di proposer LEADER.
2. Il proposer nella fase di proposa, invia questo valore agli acceptors
3. Nella fase di accept, gli acceptors si confrontano in modo tale da raggiungere tutti lo stesso stato ed eseguono la richiesta del client.
4. Nella fase di risposta viene dato l'esito al client il quale decide per maggioranza.

come sappiamo, l'algoritmo di Paxos non è resistente ai comportamenti BYZANTINI, ma bensì soltanto a fallimenti di tipo crash.

Un algoritmo in grado di resistere ai comportamenti bizantini è PBFT

# PBFT (SMA)



1. Il client invia una richiesta al server leader.
2. Il leader ritrasmette la richiesta ai nodi secondari mediante BOSSIPING, anteponendo



un numero di sequenza.

3. Nella fase di prepare, i nodi secondari si accordano con il leader e tra di loro sull'ordine delle richieste.
4. Le richieste vengono eseguite
5. Il risultato viene dato al client.

- Se il client non riceve tutte risposte uguali allora prosegue a MAGGIORANZA.  
Se il client ottiene  $f+1$  risposte identiche si raggiunge il consenso.
- Il leader viene scelto a rotazione ad ogni round dell'algoritmo oppure viene cambiato se è sospeso da parte di qualche nodo, della maliziosità del leader.

### PROBLEMI:

- ALGORITMO POCO SCALABILE (max 1000 nodi)
- N. NODI ATTIVI SEMPRE NOTO
- CAMBIO DEL LEADER SOLO IN PRECISE CIRCOSTANZE (alcuni potrebbero non darglielo mai)