MODULE *Zarb*

EXTENDS *Integers*, *Sequences*, *FiniteSets*, *TLC*

CONSTANT

The total number of faulty nodes

NumFaulty,

MaxRound

$NumValidators \triangleq (3 * NumFaulty) + 1$

$QuorumCnt \triangleq (2 * NumFaulty) + 1$

$OneThird \triangleq NumFaulty + 1$

ASSUME

$\land\ NumFaulty \geq 1$

VARIABLES

log,

states

$vars \triangleq \langle states,\ log \rangle$

Helper functions

Fetch a subset of messages in the network based on the *params* filter.

$SubsetOfMsgs(params) \triangleq$

$\{msg \in log : \forall\, field \in \text{DOMAIN } params : msg[field] = params[field]\}$

In *Zarb isProposer* is chosen based on the time a validator was joined the network
here we assume the validators joined sequentially

$IsProposer(index) \triangleq$

$(states[index].round + states[index].proposerIndex)\%NumValidators = index$

$HasPrepareQuorum(index) \triangleq$

$Cardinality(SubsetOfMsgs([$

$type \mapsto$ "PREPARE",

$height \mapsto states[index].height,$

$round \mapsto states[index].round])) \geq QuorumCnt$

$HasPrecommitQuorum(index) \triangleq$

$Cardinality(SubsetOfMsgs([$

$type \mapsto$ "PRECOMMIT",

$height \mapsto states[index].height,$

$round \mapsto states[index].round])) \geq QuorumCnt$

$HasChangeProposerQuorum(index) \triangleq$

$Cardinality(SubsetOfMsgs([$
$type \mapsto$ "CHANGE-PROPOSER",
$height \mapsto states[index].height,$
$round \mapsto states[index].round])) \geq QuorumCnt$

$HasOneThirdOfChangeProposer(index) \triangleq$
$Cardinality(SubsetOfMsgs([$
$type \mapsto$ "CHANGE-PROPOSER",
$height \mapsto states[index].height,$
$round \mapsto states[index].round])) \geq OneThird$

$GetProposal(height, round) \triangleq$
$SubsetOfMsgs([type \mapsto$ "PROPOSAL", $height \mapsto height, round \mapsto round])$

$HasProposal(height, round) \triangleq$
$Cardinality(GetProposal(height, round)) > 0$

$IsCommitted(height) \triangleq$
$Cardinality(SubsetOfMsgs([type \mapsto$ "BLOCK-ANNOUNCE", $height \mapsto height])) > 0$

SendProposal is used to broadcase proposal into the network
$SendProposal(index) \triangleq$
$log' = log \cup \{[$
$type \mapsto$ "PROPOSAL",
$height \mapsto states[index].height,$
$round \mapsto states[index].round,$
$index \mapsto index$
$]\}$

$SendPrepareVote(index) \triangleq$
$log' = log \cup \{[$
$type \mapsto$ "PREPARE",
$height \mapsto states[index].height,$
$round \mapsto states[index].round,$
$index \mapsto index$
$]\}$

$SendPrecommitVote(index) \triangleq$
$log' = log \cup \{[$
$type \mapsto$ "PRECOMMIT",
$height \mapsto states[index].height,$
$round \mapsto states[index].round,$
$index \mapsto index$
$]\}$

$SendChangeProposerRequest(index) \triangleq$
$log' = log \cup \{[$
$type \mapsto$ "CHANGE-PROPOSER",
$height \mapsto states[index].height,$
$round \mapsto states[index].round,$
$index \mapsto index$
$]\}$

$AnnounceBlock(index) \triangleq$
$log' = \{msg \in log : (msg.type =$ "BLOCK-ANNOUNCE") $\vee\ msg.height > states[index].height\} \cup \{[$
$type \mapsto$ "BLOCK-ANNOUNCE",
$height \mapsto states[index].height,$
$round \mapsto states[index].round,$
$index \mapsto -1$
$]\}$

$NewHeight(index) \triangleq$
$\wedge\ states[index].name =$ "new-height"
$\wedge\ states' = [states$ EXCEPT
$![index].name =$ "propose",
$![index].height = states[index].height + 1,$
$![index].round = 0]$
$\wedge$ UNCHANGED $\langle log \rangle$

$Propose(index) \triangleq$
$\wedge\ states[index].name =$ "propose"
$\wedge$ IF $IsProposer(index)$
THEN $SendProposal(index)$
ELSE $log' = log$
$\wedge\ states' = [states$ EXCEPT $![index].name =$ "prepare"$]$

$Prepare(index) \triangleq$
$\wedge\ states[index].name =$ "prepare"
$\wedge$ IF $\wedge\ HasProposal(states[index].height,\ states[index].round)$
$\wedge \neg HasOneThirdOfChangeProposer(index)$
$\vee\ states[index].round \geq MaxRound$
THEN $\wedge\ SendPrepareVote(index)$
$\wedge$ IF $HasPrepareQuorum(index)$
THEN $states' = [states$ EXCEPT $![index].name =$ "precommit"$]$
ELSE $states' = states$
ELSE $\wedge\ SendChangeProposerRequest(index)$

$\land\ states' = [states\ \text{EXCEPT}\ ![index].name = \text{“change-proposer”}]$

$Precommit(index)\ \triangleq$
$\land\ states[index].name = \text{“precommit”}$
$\land\ SendPrecommitVote(index)$
$\land\ \text{IF}\ HasPrecommitQuorum(index) \land \neg HasOneThirdOfChangeProposer(index)$
$\quad \text{THEN}\ states' = [states\ \text{EXCEPT}\ ![index].name = \text{“commit”}]$
$\quad \text{ELSE}\ \ states' = states$

$Commit(index)\ \triangleq$
$\land\ states[index].name = \text{“commit”}$
$\land\ AnnounceBlock(index)$
$\land\ states' = [states\ \text{EXCEPT}$
$\quad ![index].name = \text{“new-height”},$
$\quad ![index].proposerIndex = (states[index].round + 1)\%NumValidators]$

$ChangeProposer(index)\ \triangleq$
$\land\ states[index].name = \text{“change-proposer”}$
$\land\ \text{IF}\ HasChangeProposerQuorum(index)$
$\quad \text{THEN}\ states' = [states\ \text{EXCEPT}$
$\qquad ![index].name = \text{“propose”},$
$\qquad ![index].round = states[index].round + 1]$
$\quad \text{ELSE}\ \ states' = states$
$\land\ \text{UNCHANGED}\ \langle log \rangle$

$Sync(index)\ \triangleq$
$\quad \text{LET}$
$\qquad blocks\ \triangleq\ SubsetOfMsgs([type \mapsto \text{“BLOCK-ANNOUNCE”}, height \mapsto states[index].height])$
$\quad \text{IN}$
$\qquad \land\ Cardinality(blocks) > 0$
$\qquad \land\ states' = [states\ \text{EXCEPT}$
$\qquad\quad ![index].name = \text{“propose”},$
$\qquad\quad ![index].height = states[index].height + 1,$
$\qquad\quad ![index].round = 0,$
$\qquad\quad ![index].proposerIndex = ((\text{CHOOSE}\ b \in blocks : \text{TRUE}).round + 1)\%NumValidators]$
$\qquad \land\ log' = log$

$Init\ \triangleq$
$\land\ log = \{\}$
$\land\ states = [index \in 0\,..\,NumValidators - 1 \mapsto [$
$\quad name \mapsto \text{“new-height”},$
$\quad height \mapsto 0,$
$\quad round \mapsto 0,$
$\quad proposerIndex \mapsto 0$

]]

$Next \triangleq$
$\exists\, index \in 0 \,..\, NumValidators - 1 :$
$\lor\ Sync(index)$
$\lor\ NewHeight(index)$
$\lor\ Propose(index)$
$\lor\ Prepare(index)$
$\lor\ Precommit(index)$
$\lor\ Commit(index)$
$\lor\ ChangeProposer(index)$

The specification must start with the initial state and transition according to *Next*.

$Spec \triangleq$
$Init \land \Box[Next]_{vars}$

$TypeOK \triangleq$
$\land\ \ \forall\, index \in 0 \,..\, NumValidators - 1 :$
$\land\ states[index].name \in \{$"new-height", "propose", "prepare", "precommit", "commit", "change-proposer"$\}$
$\land\ \neg IsCommitted(states[index].height) \Rightarrow$
$\land\ states[index].name =$ "propose" $\Rightarrow$
$\lor\ Cardinality(SubsetOfMsgs([index \mapsto index,\ height \mapsto states[index].height,\ round \mapsto sta$
$\land\ states[index].name =$ "precommit" $\Rightarrow$
$\lor\ HasPrepareQuorum(index)$
$\land\ states[index].name =$ "commit" $\Rightarrow$
$\lor\ HasPrecommitQuorum(index)$
$\land\ \forall\, round \in 0 \,..\, states[index].round :$
$\land\ Cardinality(GetProposal(states[index].height,\ round)) \leq 1$ not more than two proposals pe
$\land\ round > 0 \Rightarrow Cardinality(SubsetOfMsgs([type \mapsto$ "CHANGE-PROPOSER", $round \mapsto ro$