```
──────────────── MODULE CassandraPaxos ────────────────
EXTENDS Integers

Maximum(S) ≜
  If S is a set of numbers, then this define Maximum(S) to be the maximum of those numbers,
  or −1 if S is empty.
  IF S = {} THEN −1
            ELSE CHOOSE n ∈ S : ∀ m ∈ S : n ≥ m

CONSTANTS Value, Acceptor, Quorum
ASSUME  ∧ ∀ Q ∈ Quorum : Q ⊆ Acceptor
        ∧ ∀ Q1, Q2 ∈ Quorum : Q1 ∩ Q2 ≠ {}

Ballot ≜ Nat

None ≜ CHOOSE v : v ∉ Value

Message ≜
       [type : {"Prepare"}, bal  : Ballot]
  ∪    [type : {"Promise"}, acc : Acceptor, bal        : Ballot,
        maxAccBal  : Ballot ∪ {−1}, maxAccVal  : Value ∪ {None},
        macComBal : Ballot ∪ {−1}, maxComVal : Value ∪ {None}]
  ∪    [type : {"Propose"}, bal : Ballot, val : Value]
  ∪    [type : {"Accept"}, acc : Acceptor, bal : Ballot, val : Value]
  ∪    [type : {"Commit"}, bal : Ballot, val : Value]
  ∪    [type : {"Ack"}, acc : Acceptor, bal : Ballot, val : Value]

VARIABLES maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal, msgs
vars ≜ ⟨maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal, msgs⟩

TypeOK ≜  ∧ maxBal    ∈ [Acceptor → Ballot ∪ {−1}]
          ∧ maxAccBal ∈ [Acceptor → Ballot  ∪ {−1}]
          ∧ maxAccVal  ∈ [Acceptor → Value  ∪ {None}]
          ∧ maxComBal ∈ [Acceptor → Ballot ∪ {−1}]
          ∧ maxComVal  ∈ [Acceptor → Value ∪ {None}]
          ∧ msgs ⊆ Message


Init ≜  ∧ maxBal   = [a ∈ Acceptor ↦ −1]
        ∧ maxAccBal = [a ∈ Acceptor ↦ −1]
        ∧ maxAccVal  = [a ∈ Acceptor ↦ None]
        ∧ maxComBal = [a ∈ Acceptor ↦ −1]
        ∧ maxComVal  = [a ∈ Acceptor ↦ None]
        ∧ msgs = {}

Send(m) ≜ msgs′ = msgs ∪ {m}


Prepare(b) ≜ ∧ Send([type ↦ "Prepare", bal ↦ b])
```

```
                    ∧ UNCHANGED ⟨maxBal, maxAccBal, maxAccVal, maxComBal,
                                 maxComVal⟩

Promise(a) ≜
  ∧ ∃ m ∈ msgs :
       ∧ m.type = "Prepare"
       ∧ m.bal > maxBal[a]
       ∧ maxBal′ = [maxBal EXCEPT ![a] = m.bal]
       ∧ Send([type ↦ "Promise", acc ↦ a, bal ↦ m.bal,
                   maxAccBal  ↦ maxAccBal[a], maxAccVal   ↦ maxAccVal[a],
                   maxComBal ↦ maxComBal[a], maxComVal ↦ maxComVal[a]])
  ∧ UNCHANGED ⟨maxAccBal, maxAccVal, maxComBal, maxComVal⟩

Propose(b, v) ≜  ∧ ¬∃ m ∈ msgs : m.type = "Propose" ∧ m.bal = b
                 ∧ ∃ Q ∈ Quorum :
                     LET Qmset ≜ {m ∈ msgs : ∧ m.type = "Promise"
                                             ∧ m.acc ∈ Q
                                             ∧ m.bal = b}
                         maxAbal ≜ Maximum({m.maxAccBal  : m ∈ Qmset})
                         maxCbal ≜ Maximum({m.maxComBal : m ∈ Qmset})
                          val ≜ IF maxAbal > maxCbal
                                  THEN (CHOOSE m ∈ Qmset : m.maxAccBal = maxAbal).maxAccVal
                                  ELSE  v
                     IN     ∧ ∀ a ∈ Q : ∃ m ∈ Qmset : m.acc = a
                            ∧ Send([type ↦ "Propose", bal ↦ b, val ↦ val])
                 ∧ UNCHANGED ⟨maxBal, maxAccBal, maxAccVal, maxComBal, maxComVal⟩


Accept(a) ≜  ∧ ∃ m ∈ msgs : ∧ m.type = "Propose"
                            ∧ maxBal[a] ≤ m.bal
                            ∧ maxBal′ = [maxBal EXCEPT ![a] = m.bal]
                            ∧ maxAccBal′ = [maxAccBal EXCEPT ![a] = m.bal]
                            ∧ maxAccVal′ = [maxAccVal EXCEPT ![a] = m.val]
                            ∧ Send([type ↦ "Accept", bal ↦ m.bal, val ↦ m.val,
                                     acc ↦ a])
             ∧ UNCHANGED ⟨maxComBal, maxComVal⟩

Commit(b, v) ≜  ∧ ¬∃ m ∈ msgs : m.type = "Commit" ∧ m.bal = b
                ∧ ∃ Q ∈ Quorum :
                    LET QAmset ≜ {m ∈ msgs : ∧ m.type = "Accept"
                                             ∧ m.acc ∈ Q
                                             ∧ m.bal = b}
                    IN   ∧ ∀ a ∈ Q : ∃ m ∈ QAmset : m.acc = a
                ∧ Send([type ↦ "Commit", bal ↦ b, val ↦ v])
```

```
                         $\land$ UNCHANGED $\langle maxBal, maxAccBal, maxAccVal, maxComBal,$
                                           $maxComVal \rangle$

$Ack(a) \triangleq$  $\land \exists m \in msgs :$ $\land m.type =$ "Commit"
                                        $\land maxBal[a] \leq m.bal$
                                        $\land maxBal' = [maxBal$ EXCEPT $![a] = m.bal]$
                                        $\land maxComBal' = [maxComBal$ EXCEPT $![a] = m.bal]$
                                        $\land maxComVal' = [maxComVal$ EXCEPT $![a] = m.val]$
                                        $\land Send([type \mapsto$ "Ack", $bal \mapsto m.bal, val \mapsto m.val,$
                                                $acc \mapsto a])$
                     $\land$ UNCHANGED $\langle maxAccBal, maxAccVal \rangle$

$Next \triangleq$  $\lor \exists b \in Ballot :$ $\lor Prepare(b)$
                                   $\lor \exists v \in Value : Propose(b, v) \lor Commit(b, v)$
               $\lor \exists a \in Acceptor :$ $\lor Promise(a) \lor Accept(a) \lor Ack(a)$

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

* Modification History

* Last modified *Thu Dec* 09 19:33:06 *CST* 2021 by *LENOVO*

* Created *Thu Dec* 02 10:19:29 *CST* 2021 by *LENOVO*