```
1  ---------- MODULE TCS ----------
   See DISC'2018: Multi-Shot Distributed Transaction Commit
5  EXTENDS Naturals, Integers, FiniteSets, Sequences, Functions, TLC
6  ----------
7  CONSTANTS
8      Key,          the set of keys, ranged over by k ∈ Key
9      Tid,          the set of transaction identifiers, ranged over by t ∈ Tid
10     RSet,         RSet[t]: the read set of t ∈ Tid
11     WSet,         WSet[t]: the write set of t ∈ Tid
12     CVer,         CVer[t]: the commit version of t ∈ Tid
13     Shard,        the set of shards, ranged over by s ∈ Shard
14     Coord,        Coord[t]: the coordinator of t ∈ Tid
15     KeySharding   KeySharding[k]: the shard that holds k ∈ Key

17 NotTid ≜ CHOOSE t : t ∉ Tid

19 Ver ≜ 0 .. Cardinality(Tid)  with a distinguished minimum version 0
20 Slot ≜ 0 .. Cardinality(Tid) − 1

22 TKey(t) ≜ WSet[t] ∪ {kv[1] : kv ∈ RSet[t]}
23 TSharding(t) ≜ {KeySharding[k] : k ∈ TKey(t)}

25 ASSUME
26     ∧ RSet ∈ [Tid → SUBSET (Key × Ver)]
27     ∧ ∀ t ∈ Tid: RSet[t] \* TODO: one version per object
28     ∧ WSet ∈ [Tid → SUBSET Key]
29     ∧ \* TODO: "no blind update" assumption
30     ∧ CVer ∈ [Tid → Ver]
31     ∧ \* TODO: higher than any of the versions read
32     ∧ Coord ∈ [Tid → Shard]
33     ∧ KeySharding ∈ [Key → Shard]
34 ----------
35 VARIABLES
36     next,      next[s] ∈ Z points to the last filled slot
37     txn,       txn[s][i] is the transaction (identifier) to certify in the i-th slot
38     vote,      vote[s][i] is the vote for txn[s][i]
39     dec,       dec[s][i] is the decision for txn[s][i]
40     phase,     phase[s][i] is the phase for txn[s][i]
41     msg,       the set of messages in transit
42     submitted  the set of t ∈ Tid that have been submitted to TCS

44 sVars ≜ ⟨next, txn, vote, dec, phase⟩
45 vars ≜ ⟨next, txn, vote, dec, phase, msg, submitted⟩
46 ----------
47 Message ≜ [type : {"PREPARE"}, t : Tid, s : Shard]
48     ∪ [type : {"PREPARE_ACK"}, s : Shard, n : Int, t : Tid, v : {"COMMIT", "ABORT"}]
```

```
49      ∪ [type : {"DECISION"}, pos : Int, d : {"COMMIT", "ABORT"}, s : Shard]

51  Send(m) ≜ msg' = msg ∪ m
52  Delete(m) ≜ msg' = msg \ m
53  SendAndDelete(sm, dm) ≜ msg' = (msg ∪ sm) \ dm
54  ----------------------------------------------------------------------
55  TypeOK ≜
56      ∧  next ∈ [Shard → Int]
57      ∧  txn ∈ [Shard → [Slot → Tid ∪ {NotTid}]]
58      ∧  vote ∈ [Shard → [Slot → {"COMMIT", "ABORT", "NULL"}]]
59      ∧  dec ∈ [Shard → [Slot → {"COMMIT", "ABORT", "NULL"}]]
60      ∧  phase ∈ [Shard → [Slot → {"START", "PREPARED", "DECIDED"}]]
61      ∧  msg ⊆ Message
62      ∧  submitted ⊆ Tid
63  ----------------------------------------------------------------------
64  Init ≜
65      ∧ next = [s ∈ Shard ↦ −1]
66      ∧ txn = [s ∈ Shard ↦ [i ∈ Slot ↦ NotTid]]
67      ∧ vote = [s ∈ Shard ↦ [i ∈ Slot ↦ "NULL"]]
68      ∧ dec = [s ∈ Shard ↦ [i ∈ Slot ↦ "NULL"]]
69      ∧ phase = [s ∈ Shard ↦ [i ∈ Slot ↦ "START"]]
70      ∧ msg = {}
71      ∧ submitted = {}
72  ----------------------------------------------------------------------
73  Vote(t, s, n) ≜ "ABORT"  TODO
74  ----------------------------------------------------------------------
75  Certify(t) ≜  Certify t ∈ Tid
76      ∧  t ∈ Tid \ submitted
77      ∧  Send([type : {"PREPARE"}, t : {t}, s : TSharding(t)])
78      ∧  submitted' = submitted ∪ {t}
79      ∧  UNCHANGED sVars

81  Prepare(t, s) ≜  Prepare t ∈ Tid on s ∈ Shard when receive "PREPARE(t)" message
82      ∧  ∃ m ∈ msg :
83          ∧ m = [type ↦ "PREPARE", t ↦ t, s ↦ s]
84          ∧ next' = [next EXCEPT ![s] = @ + 1]
85          ∧ txn' = [txn EXCEPT ![s][next'[s]] = t]
86          ∧ vote' = [vote EXCEPT ![s][next'[s]] = Vote(t, s, next'[s])]  TODO
87          ∧ phase' = [phase EXCEPT ![s][next'[s]] = "PREPARED"]
88          ∧ SendAndDelete({[type ↦ "PREPARE_ACK", s ↦ s, n ↦ next'[s],
89                            t ↦ t, v ↦ vote'[s][next'[s]]]}, {m})
90      ∧  UNCHANGED ⟨dec, submitted⟩
91  ----------------------------------------------------------------------
92  Next ≜
93      ∨ ∃ t ∈ Tid : Certify(t)
```

$$\lor \exists\, t \in \mathit{Tid},\ s \in \mathit{Shard} : \mathit{Prepare}(t,\ s)$$

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

* Modification History
* Last modified Sun *Jun* 13 11:30:53 *CST* 2021 by *hengxin*
* Created Sat *Jun* 12 21:01:57 *CST* 2021 by *hengxin*