```
┌──────────────────────── MODULE Skeen ────────────────────────┐
The specification of Skeen's protocol for atomic multicast; see Section III of the DSN2019 paper
"White-Box Atomic Multicast" by Alexey Gotsman, Anatole Lefort, and Gregory Chockler.
EXTENDS Naturals, Sequences, FiniteSets, TLC
├──────────────────────────────────────────────────────────────┤
Injective(f) ≜ ∀ a, b ∈ DOMAIN f : (a ≠ b) ⇒ (f[a] ≠ f[b])

Max(a, b) ≜ IF a > b THEN a ELSE b
├──────────────────────────────────────────────────────────────┤
CONSTANTS
    Msg,      the set of messages, ranged over by m
    Proc,     the set of processes, ranged over by p
    Dest      Dest[m] ⊆ Proc: the set of destination processes of m ∈ Msg

ASSUME
    ∧ Dest ∈ [Msg → SUBSET Proc]

Priority ≜ CHOOSE f ∈ [Proc → 1 . . Cardinality(Proc)] : Injective(f)
├──────────────────────────────────────────────────────────────┤
VARIABLES
    clock,        clock[p]: the clock at process p ∈ Proc
    phase,        phase[p][m]: the phase of the message m ∈ Msg at process p ∈ Proc
    localTS,      localTS[p][m]: the local ts of the message m ∈ Msg at process p ∈ Proc
    globalTS,     globalTS[p][m]: the global ts of the message m ∈ Msg at process p ∈ Proc
    delivered,    delivered[p][m]: has m ∈ Msg been delivered at process p ∈ Proc
    incoming,      incoming[p] ⊆ Message (defined below): the incoming channel of process p ∈ Proc
    sent          sent ⊆ Msg: the set of messages that have been multicast; only for TLC

pvars ≜ ⟨clock, phase, localTS, globalTS, delivered⟩
vars ≜ ⟨clock, phase, localTS, globalTS, delivered, incoming, sent⟩
├──────────────────────────────────────────────────────────────┤
TS ≜ [c : 0 . . Cardinality(Msg), p : Proc]  c for counter

GT(u, v) ≜  Is u > v?
      ∨ u.c > v.c
      ∨ ∧ u.c = v.c
        ∧ Priority[u.p] > Priority[v.p]

MaxV(vs) ≜ CHOOSE u ∈ vs : ∀ v ∈ vs : u ≠ v ⇒ GT(u, v)
├──────────────────────────────────────────────────────────────┤
Message ≜ [type : {"MULTICAST"}, m : Msg]
    ∪ [type : {"PROPOSE"}, m : Msg, p : Proc, lts : TS]

Send(msg) ≜  Send msg ∈ Message to its destination processes
    incoming′ = [p ∈ Proc ↦
        IF p ∈ Dest[msg.m] THEN incoming[p] ∪ {msg}
                           ELSE incoming[p]]
```

Send smsg ∈ Message to its destination processes and remove rmsg ∈ Message from incoming[sender]

Precondition: sender ∈ Dest[msg.m]

```
SendAndRemove(smsg, sender, rmsg) ≜
    incoming′ = [p ∈ Proc ↦
        IF p = sender THEN (incoming[sender] ∪ {smsg}) \ {rmsg}
                      ELSE IF p ∈ Dest[smsg.m] THEN incoming[p] ∪ {smsg}
                                               ELSE incoming[p]]
────────────────────────────────────────────────────────────────────────
TypeOK ≜
    ∧ clock    ∈ [Proc → 0 . . Cardinality(Msg)]
    ∧ phase    ∈ [Proc → [Msg → {"START", "PROPOSED", "COMMITTED"}]]
    ∧ localTS  ∈ [Proc → [Msg → TS]]
    ∧ globalTS ∈ [Proc → [Msg → TS]]
    ∧ delivered ∈ [Proc → [Msg → BOOLEAN ]]
    ∧ incoming ∈ [Proc → SUBSET Message]
    ∧ sent     ⊆ Msg
────────────────────────────────────────────────────────────────────────
Init ≜
    ∧ clock    = [p ∈ Proc ↦ 0]
    ∧ phase    = [p ∈ Proc ↦ [m ∈ Msg ↦ "START"]]
    ∧ localTS  = [p ∈ Proc ↦ [m ∈ Msg ↦ [c ↦ 0, p ↦ p]]]
    ∧ globalTS = [p ∈ Proc ↦ [m ∈ Msg ↦ [c ↦ 0, p ↦ p]]]
    ∧ delivered = [p ∈ Proc ↦ [m ∈ Msg ↦ FALSE]]
    ∧ incoming  = [p ∈ Proc ↦ {}]
    ∧ sent     = {}
────────────────────────────────────────────────────────────────────────
Multicast(m) ≜   Multicast m ∈ Msg
    ∧ m ∈ Msg \ sent
    ∧ sent′ = sent ∪ {m}
    ∧ Send([type ↦ "MULTICAST", m ↦ m])
    ∧ UNCHANGED pvars

Propose(p) ≜   When p ∈ Proc receives a MULTICAST for some m ∈ Msg
    ∃ msg ∈ incoming[p] :
        ∧ msg.type = "MULTICAST"
        ∧ LET m ≜ msg.m
          IN  ∧ Assert(p ∈ Dest[m], "p should be one of the destination process of m")
              ∧ clock′ = [clock EXCEPT ![p] = @ + 1]
              ∧ localTS′ = [localTS EXCEPT ![p][m] = [c ↦ clock′[p], p ↦ p]]
              ∧ phase′ = [phase EXCEPT ![p][m] = "PROPOSED"]
              ∧ SendAndRemove([type ↦ "PROPOSE", m ↦ m, p ↦ p,
                               lts  ↦ localTS′[p][m]], p, msg)
              ∧ UNCHANGED ⟨globalTS, delivered, sent⟩
```

```
$Deliver(p) \triangleq$   When $p \in Proc$ receives all $PROPOSE$ for some $m \in Msg$
    $\exists\, m \in Msg :$
        LET $msgofm \triangleq \{msg \in incoming[p] : msg.type = \text{“PROPOSE”} \land msg.m = m\}$
              $destofm \triangleq \{msg.p : msg \in msgofm\}$
              $ltsofm \triangleq \{msg.lts : msg \in msgofm\}$
        IN    $\land\ destofm = Dest[m]$
              $\land\ globalTS' = [globalTS$ EXCEPT $![p][m] = MaxV(ltsofm)]$
              $\land\ clock' = [clock$ EXCEPT $![p] = Max(clock[p], globalTS'[p][m].c)]$
              $\land\ phase' = [phase$ EXCEPT $![p][m] = \text{“COMMITTED”}]$
              $\land$ LET $readym \triangleq \{rm \in Msg :$
                                 $\land\ phase'[p][rm] = \text{“COMMITTED”}$
                                 $\land\ delivered[p][rm] =$ FALSE
                                 $\land\ \forall\, pm \in Msg :$
                                     $phase'[p][pm] = \text{“PROPOSED”}$
                                         $\Rightarrow GT(localTS[p][pm], globalTS'[p][rm])\}$
                  IN  $delivered' = [delivered$ EXCEPT $![p] = [pm \in Msg \mapsto$
                                 IF $pm \in readym$ THEN TRUE ELSE $@[pm]]]$
              $\land$ UNCHANGED $\langle localTS, sent, incoming \rangle$
--------------------------------------------------------------------------------
$Next \triangleq$
    $\lor\ \exists\, m \in Msg : Multicast(m)$
    $\lor\ \exists\, p \in Proc :$
        $\lor\ Propose(p)$
        $\lor\ Deliver(p)$

$Spec \triangleq Init \land \Box[Next]_{vars}$
--------------------------------------------------------------------------------
Invariant: Global timestamps are unique for each $m \in Msg$; see Section $III$.
$UniqueGTS \triangleq$
    $\forall\, p \in Proc, m1, m2 \in Msg :$
        $(m1 \neq m2 \land phase[p][m1] = \text{“COMMITTED”} \land phase[p][m2] = \text{“COMMITTED”})$
            $\Rightarrow globalTS[p][m1] \neq globalTS[p][m2]$

Invariant: Each $m \in Msg$ is assigned a single global timestamp.
$SameGTS \triangleq$
    $\forall\, p1, p2 \in Proc, m \in Msg :$
        $(phase[p1][m] = \text{“COMMITTED”} \land phase[p2][m] = \text{“COMMITTED”})$
            $\Rightarrow globalTS[p1][m] = globalTS[p2][m]$
--------------------------------------------------------------------------------
THEOREM $TypeTheorem \triangleq Spec \Rightarrow \Box TypeOK$

THEOREM $UniqueGTSTheorem \triangleq Spec \Rightarrow \Box UniqueGTS$

THEOREM $SameGTSTheorem \triangleq Spec \Rightarrow \Box SameGTS$
================================================================================
\* Modification History
```

\ * Last modified *Fri Jul* 23 20:42:57 *CST* 2021 by *hengxin*
\ * Last modified *Thu Jul* 08 20:18:10 *CST* 2021 by *eric*
\ * Created Sat *Jun* 26 16:04:39 *CST* 2021 by *eric*