```
1  ---------- MODULE Leader ----------
2  EXTENDS Reals, Bags
3  ----------------------------------
4  CONSTANTS
5      N,          number of nodes
6      Nbrs(_),    Nbrs(n): the neighbors of node n
7      Period,     timer for nodes that believe themselves to be the leader
8      MsgDelay,   A message is received at most MsgDelay seconds after it is sent
9      TODelay     Nodes can be awakened up to TODelay seconds after timeout

11 Node ≜ 1 .. N

13 ASSUME
14     ∧ N ∈ Nat
15     ∧ ∀ n ∈ Node :
16         ∧ Nbrs(n) ⊆ Node
17         ∧ ∀ m ∈ Nbrs(n) : n ∈ Nbrs(m)
18     ∧ {Period, MsgDelay, TODelay} ⊆ {r ∈ Real : r > 0}
19 ----------------------------------
20 VARIABLES
21     ldr,        ldr[n]: The node that n believes to be its leader.
22     dist,       dist[n]: What n believes to be its distance to ldr[n].
23     timer,      A countdown timer for node n's timeout action.
24     msgs,       the messages in transit; there may be duplicate messages
25     now         the now timer

27 vars ≜ ⟨ldr, dist, timer, msgs, now⟩

29 Msg ≜ [src   : Node,   the sender
30        dest  : Node,   the destination
31        ldr   : Node,   the leader that originated the message
32        hops  : Nat,    the number of times the message has been forwarded
33    rcvTimer  : Real]   a countdown timer used to express the upper-bound constraint on message-delivery time
34 ----------------------------------
35 TypeOK ≜
36     ∧  ldr ∈ Node
37     ∧  dist ∈ Nat
38     ∧  timer ∈ Real
39     ∧  msgs ∈ SubBag(SetToBag(Msg))
40     ∧  now ∈ Real    now is unbounded; commet it in model checking
41 ----------------------------------
42 Init ≜
43     ∧ ldr   = [n ∈ Node ↦ n]
44     ∧ dist  = [n ∈ Node ↦ 0]
45     ∧ timer = [n ∈ Node ↦ Period]
46     ∧ msgs  = EmptyBag
```

```
47      ∧ now   = 0
48  ----------------------------------------------------------------------------
49  MsgsSent(n, S) ≜
50      SetToBag([src  : {n}, dest : S, ldr : {ldr'[n]},
51                hops : {dist'[n]}, rcvTimer : {MsgDelay}])

53  TimeOut(n) ≜
54      ∧    timer[n] < 0
55      ∧    ldr'    = [ldr EXCEPT ![n] = n]
56      ∧    dist'   = [dist EXCEPT ![n] = 0]
57      ∧    timer'  = [timer EXCEPT ![n] = Period]
58      ∧    msgs'   = msgs ⊕ MsgsSent(n, Nbrs(n))
59      ∧    UNCHANGED now

61  RcvMsg(n) ≜
62      ∧ ∃ m ∈ BagToSet(msgs) :
63          ∧ m.dest = n
64          ∧ IF  ∨ m.ldr < ldr[n]
65                ∨ ∧ m.ldr = ldr[n]
66                  ∧ m.hops + 1 < dist[n]   TODO: " ≤ " in Lamport's spec?
67            THEN  ∧ ldr'  = [ldr EXCEPT ![n] = m.ldr]
68                  ∧ dist' = [dist EXCEPT ![n] = m.hops + 1]
69                  ∧ timer' = [timer EXCEPT ![n] =
70                                  Period + TODelay + dist'[n] * MsgDelay]
71                  ∧ msgs' = (msgs ⊖ SetToBag({m}))
72                                  ⊕ MsgsSent(n, Nbrs(n) \ {m.src})
73            ELSE  ∧ msgs' = msgs ⊖ SetToBag({m})
74                  ∧ UNCHANGED ⟨ldr, dist, timer⟩
75      ∧ UNCHANGED now

77  Tick ≜
78      ∃ d ∈ Real :
79          ∧ d > 0
80          ∧ ∀ n  ∈ Node : timer[n] + TODelay ≥ d
81          ∧ ∀ m ∈ BagToSet(msgs) : m.rcvTimder ≥ d
82          ∧ timer' = [n ∈ Node ↦ timer[n] − d]
83          ∧ msgs' = LET Updated(m) ≜ [m EXCEPT !.rcvTimer = @ − d]
84                    IN   BagOfAll(Updated, msgs)
85          ∧ now' = now + d
86          ∧ UNCHANGED ⟨ldr, dist⟩

88  Next ≜
89      ∨ ∃ n ∈ Node : TimeOut(n) ∨ RcvMsg(n)
90      ∨ Tick
91  ----------------------------------------------------------------------------
92  LSpec ≜ Init ∧ □[Next]_vars
```

```
93  ----------------------------------------------------------------------------
94  Min(S) ≜ CHOOSE i ∈ S : ∀ j ∈ S : i ≤ j

96  Ball(i, n) ≜  The set of nodes with distance of at most i from node n.
97      LET B[j ∈ 0 .. i] ≜
98          IF j = 0 THEN {n}
99                   ELSE  B[j − 1] ∪ UNION {Nbrs(m) : m ∈ B[j − 1]}
100     IN   B[i]

102 Dist(m, n) ≜  The distance between nodes m and n, if it is finite.
103     Min({i ∈ 0 .. N : m ∈ Ball(i, n)})
104 ----------------------------------------------------------------------------
105 Correctness ≜
106     LET Ldr(n) ≜ Min(Ball(N, n))
107     IN  ∀ n ∈ Node :
108             (now > Period + TODelay + Dist(n, Ldr(n) ∗ MsgDelay)
109                 ⇒ ldr[n] = Ldr(n))

111 THEOREM LSpec ⇒ □Correctness
112 ----------------------------------------------------------------------------
    \ ∗ Modification History
    \ ∗ Last modified Fri Aug 06 14:08:46 CST 2021 by hengxin
    \ ∗ Created Fri Aug 06 11:55:18 CST 2021 by hengxin
```