```
MODULE BestEffortBroadcast

EXTENDS
    Naturals,
    FiniteSets,
    Bags

CONSTANTS
    Procs,
    Messages,
    Correct

ASSUME
    ∧ Procs ≠ {}
    ∧ Messages ≠ {}
    ∧ Correct ∈ SUBSET Procs

BC_Message ≜ [sdr : Procs, msg : Messages]

Let's import the perfect point-to-point links spec
See the PerfectPointToPointLink module for more details

> "I have observed that many new users want to write TLA+ specs so they
> can be reused. I have one word of advice for those users: Don't."
> https://groups.google.com/g/tlaplus/c/BHBNTkJ2QFE/m/meTQs4pHBwAJ

VARIABLES
    pl_sent,
    pl_delivered

pl_vars ≜ ⟨pl_sent, pl_delivered⟩

PL_Rich_Message ≜ [sdr : Procs, rcv : Procs, msg : BC_Message]
PL_Sent ≜ PL_Rich_Message
PL_Delivered ≜ PL_Rich_Message

pl_bcast_send(p, qs, m) ≜
    ∧ p ∈ Procs
    ∧ qs ⊆ Procs
    ∧ LET rms ≜ {[sdr ↦ p, rcv ↦ q, msg ↦ m] : q ∈ qs}
          IN
           ∧ ∀ rm ∈ rms : rm ∉ pl_sent
           ∧ pl_sent' = pl_sent ∪ rms
           ∧ UNCHANGED pl_delivered

pl_send(p, q, m) ≜
    ∧   p ∈ Procs
```

```
50      ∧  q ∈ Procs
51      ∧  LET rm ≜ [sdr ↦ p, rcv ↦ q, msg ↦ m]
52           IN
53            ∧ rm ∉ pl_sent
54            ∧ pl_sent′ = pl_sent ∪ {rm}
55            ∧ UNCHANGED pl_delivered

57  pl_deliver(p, q, m) ≜
58      ∧ p ∈ Procs
59      ∧ q ∈ Procs
60      ∧ LET rm ≜ [sdr ↦ p, rcv ↦ q, msg ↦ m]
61          IN
62           ∧ rm ∈ pl_sent
63           ∧ rm ∉ pl_delivered
64           ∧ pl_delivered′ = pl_delivered ∪ {rm}
65           ∧ UNCHANGED pl_sent

67  PL_Init ≜
68      ∧ pl_sent = {}
69      ∧ pl_delivered = {}

71  ----------------------------------------------------------------------------

73   Back to the best-effort broadcast module

75  VARIABLES
76      bc_sent,
77      bc_delivered,
78      bc_failed,
79      bc_messages_used

81  bc_vars ≜ ⟨bc_sent, bc_delivered, bc_failed, bc_messages_used⟩

83  vars ≜ ⟨pl_vars, bc_vars⟩

85  ----------------------------------------------------------------------------

87   broadcast message m from process p
88  beb_broadcast(p, m) ≜
89      ∧ m ∉ bc_messages_used
90      ∧ bc_messages_used′ = bc_messages_used ∪ {m}
91      ∧ p ∉ bc_failed
92      ∧ LET qs ≜ ProcsIN
93          LET bc_msg ≜ [sdr ↦ p, msg ↦ m]
94              IN
95              pl_bcast_send(p, qs, bc_msg)
96      ∧ bc_sent′ = bc_sent ⊕ SetToBag({[sdr ↦ p, rcv ↦ q, msg ↦ m] : q ∈ Procs})
97      ∧ UNCHANGED ⟨bc_delivered, bc_failed⟩
```

```
99   deliver a broadcast message m to process p from process q
100  beb_deliver(p, q, m) ≜
101      ∧ p ∉ bc_failed
102      ∧ LET bc_msg ≜ [sdr ↦ q, msg ↦ m]
103            IN
104            pl_deliver(q, p, bc_msg)
105      ∧ bc_delivered' = bc_delivered ⊕ SetToBag({[sdr ↦ q, rcv ↦ p, msg ↦ m]})
106      ∧ UNCHANGED ⟨bc_sent, bc_failed, bc_messages_used⟩

108  beb_fail(p) ≜
109      ∧ p ∉ Correct
110      ∧ bc_failed' = bc_failed ∪ {p}
111      ∧ UNCHANGED ⟨pl_vars, bc_sent, bc_delivered, bc_messages_used⟩

113  BEB_Init ≜
114      ∧ bc_sent = EmptyBag
115      ∧ bc_delivered = EmptyBag
116      ∧ bc_failed = {}
117      ∧ bc_messages_used = {}

119  Init ≜
120      ∧ PL_Init
121      ∧ BEB_Init

123  Next ≜ ∃ p ∈ Procs, q ∈ Procs, m ∈ Messages :
124      ∨ beb_broadcast(p, m)
125      ∨ beb_deliver(p, q, m)
126      ∨ beb_fail(p)

128  Spec ≜
129      ∧ Init
130      ∧ □[Next]_vars
131      ∧ WF_vars(Next)

133  ⊢————————————————————————————————————————————————————————————⊣

135   Let's check some properties with TLC

137  TypeInv ≜
138      ∧ pl_sent ⊆ PL_Sent
139      ∧ pl_delivered ⊆ PL_Delivered

141   BEB1: Validity: If a correct process broadcasts a message m, then every correct
142   process eventually delivers m.
143  Prop_BEB1_Validity ≜
144      □∀ p  ∈ Procs, q ∈ Procs, m ∈ Messages :
145          (p ∈ Correct ∧ q ∈ Correct) ⇒
146              (([sdr ↦ p, rcv ↦ q, msg ↦ m] ∈ DOMAIN bc_sent) ⇒
```

```
147                 (◇([sdr ↦ p, rcv ↦ q, msg ↦ m] ∈ DOMAIN bc_delivered)))

149   BEB2: No duplication: No message is delivered more than once.
150 Prop_BEB2_NoDuplication ≜
151     □∀ m ∈ BagToSet(bc_delivered) :
152         (IF BagIn(m, bc_delivered) THEN bc_delivered[m] ELSE 0) ≤ 1
153           (CopiesIn(m, bc_delivered) ≤ 1) \* This doesn't work on the Toolbox, but works in VS Code

155   BEB3: No creation: If a process delivers a message m with sender s, then m was
156   previously broadcast by process s.
157 Prop_BEB3_NoCreation ≜ □(BagToSet(bc_delivered) ⊆ BagToSet(bc_sent))

159 ╘═══════════════════════════════════════════════════════════════════════════

    \* Modification History
    \* Last modified Thu Oct 10 14:28:19 CEST 2024 by jonasspenger
    \* Created Wed Oct 09 10:21:00 CEST 2024 by jonasspenger
```