MODULE *ChannelB*

EXTENDS *Bags*, *FiniteSets*, *Naturals*, *TLC*

CONSTANTS
- *Message*, — The set of messages that can be sent and received on this channel.
- *MaxDuplicates* — The maximum number of duplicate messages allowed on the network at any one point in time.

VARIABLES
- *sent*, — The bag of sent messages.
- *network*, — The bag of messages on the network waiting to be received.
- *received* — The set of received messages.

vars $\triangleq \langle sent, network, received \rangle$ — The tuple of all variables.

The initial state. The sent messages and network are declared as empty bags. The set of received messages is initialised to the empty set.

$$Init \triangleq \begin{array}{l} \land\ sent = EmptyBag \\ \land\ network = EmptyBag \\ \land\ received = \{\} \end{array}$$

The send message action adds the message to the bag of sent messages and the network, not changing the set of received messages.

$$Send(msg) \triangleq \begin{array}{l} \land\ sent' = sent \oplus SetToBag(\{msg\}) \\ \land\ network' = network \oplus SetToBag(\{msg\}) \\ \land\ \text{UNCHANGED}\ \langle received \rangle \end{array}$$

The receive message action. If the message is present on the network add the message to the set of received messages and take the message from the network not changing the bag of sent messages.

$$Receive(msg) \triangleq \begin{array}{l} \land\ BagIn(msg, network) \\ \land\ received' = received \cup \{msg\} \\ \land\ network' = network \ominus SetToBag(\{msg\}) \\ \land\ \text{UNCHANGED}\ \langle sent \rangle \end{array}$$

The next state is found through either sending or receiving a message.

$$Next \triangleq \exists\, msg \in Message :\ Send(msg) \lor Receive(msg)$$

The type invariants for the specification. Sent and network are bags and only contain items from the set *Message*. The received set is finite and can only contain items from the set *Message*.

$$TypeInvariant \triangleq \begin{array}{l} \land\ IsABag(sent) \\ \land\ BagToSet(sent) \subseteq Message \\ \land\ IsABag(network) \end{array}$$

$$\begin{array}{l} \land\ BagToSet(network) \subseteq Message \\ \land\ IsFiniteSet(received) \\ \land\ received \subseteq Message \end{array}$$

The liveness property is that the network is eventually empty.

$$Liveness \triangleq \Diamond(network = EmptyBag)$$

The safety properties for the specification are that all messages on the network and all received messages must have been sent.

$$\begin{array}{ll} Safety & \triangleq \land \Box(network \sqsubseteq sent) \\ & \quad \land \Box(received \subseteq BagToSet(sent)) \end{array}$$

The message constraint allows a maximum number of duplicate messages on the network at any one point in time. Without this constraint checking the model would take forever - it would never complete.

$$MsgConstraint \triangleq \forall\, msg \in Message : CopiesIn(msg,\, sent) \leq MaxDuplicates$$

The channels specification and theorems.

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

THEOREM $ChannelBSpec \Rightarrow \Box\, TypeInvariant$
THEOREM $ChannelBSpec \Rightarrow Liveness \land Safety$