PERSISTENT MESSAGING PROTOCOL

The following refinement solves an issue regarding the timeout mechanism in the persistent messaging protocol discussed in *Silberschatz et al., Database System Concepts, 6th ed., 2011.*

Components

Sender $\mathcal{S}$
Receiver $\mathcal{R}$
} we allow temporary failure

Temporary failures make a node unreachable (e.g., because of network or node failure) for a finite amount of time. After the failure, the node returns and continues processing messages.

Messaging Infrastructure

- arbitrary delay (i.e., messages may get lost or may not arrive in sending order)
- probability of successful delivery > 0 (i.e., not all messages get lost)

Protocol

All additions to the original protocol (described in *Silberschatz et al.*) are marked in red.

MTS(i, M_i, t_i): *messages_to_send* relation
RM(i, M_i, t_i): *received_messages* relation
t_i: time when message M_i with unique message number i was first sent by $\mathcal{S}$

$\mathcal{S}$

- `for each` $(i, M_i, t_i) \in$ `MTS:`
  `send` (i, M_i, t_i) `to` $\mathcal{R}$
- $(i,$`acknowledge) received:`
  `MTS = MTS` $\setminus \{(j, M_j, t_j) \mid (j, M_j, t_j) \in MTS, j = i\}$
- `do periodically:`
  $T_{OLD} = min\{t_i \mid (i, M_i, t_i) \in$ `MTS`$\}$
  `send` T_{OLD} `to` $\mathcal{R}$

$\mathcal{R}$

- (i, M_i, t_i) `received:`
  `if` $t_i \geq T_{OLD}$ `:`
  `if` $(i, M_i, t_i) \notin$ `RM:`
  `RM = RM` $\cup \{(i, M_i, t_i)\}$
  `send` $(i,$`acknowledge) to` $\mathcal{S}$
- T_{OLD} `received:`
  `RM = RM` $\setminus \{(i, M_i, t_i) \mid (i, M_i, t_i) \in RM, t_i < T_{OLD}\}$