Algorithm – Uniform Broadcast

Algorithm 1 All-Ack Uniform Reliable Broadcast

Implements:

UniformReliableBroadcast, **instance** *urb*.

Uses:

BestEffortBroadcast, **instance** *beb*.

PerfectFailureDetector, **instance** P.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $\quad$ *delivered* $:= \emptyset$
3: $\quad$ *pending* $:= \emptyset$
4: $\quad$ *correct* $:= \Pi$
5: $\quad$ **for all** m **do** $\quad \triangleright$ Where m is a meta variable for a message
6: $\quad\quad$ *ack*$[m] := \emptyset$
7: **upon event** $\langle$ *urb*, *Broadcast* | m $\rangle$ **do**
8: $\quad$ *pending* $:=$ *pending* $\cup \{(self, m)\}$
9: $\quad$ **trigger** $\langle$ *beb*, *Broadcast* | [DATA, *self*, m] $\rangle$
10: **upon event** $\langle$ *beb*, *Deliver* | p, [DATA, s, m] $\rangle$ **do**
11: $\quad$ *ack*$[m] :=$ *ack*$[m] \cup \{p\}$
12: $\quad$ **if** $(s, m) \notin$ *pending* **then**
13: $\quad\quad$ *pending* $:=$ *pending* $\cup \{(s, m)\}$
14: $\quad\quad$ **trigger** $\langle$ *beb*, *Broadcast* | [DATA, s, m] $\rangle$
15: **upon event** $\langle$ P, *Crash* | p $\rangle$ **do**
16: $\quad$ *correct* $:=$ *correct* $\setminus \{p\}$
17: **function** SHOULDDELIVER(m)
18: $\quad$ **return** *correct* $\subseteq$ *ack*$[m] \wedge m \notin$ *delivered*
19: **upon event** $\langle$ $\exists_{(s,m) \in pending}$SHOULDDELIVER(m) $\rangle$ **do**
20: $\quad$ *delivered* $:=$ *delivered* $\cup \{m\}$
21: $\quad$ **trigger** $\langle$ *urb*, *Deliver* | s, m $\rangle$

Algorithm 2 Majority-Ack Uniform Reliable Broadcast

Implements:

UniformReliableBroadcast, **instance** *urb*.

Uses:

BestEffortBroadcast, **instance** *beb*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $delivered := \emptyset$
3: $pending := \emptyset$
4: $correct := \Pi$
5: $N := |\Pi|$
6: **for all** m **do** ▷ Where m is a meta variable for a message
7: $ack[m] := \emptyset$
8: **upon event** $\langle$ *urb*, *Broadcast* | m $\rangle$ **do**
9: $pending := pending \cup \{(self, m)\}$
10: **trigger** $\langle$ *beb*, *Broadcast* | [DATA, *self*, m] $\rangle$
11: **upon event** $\langle$ *beb*, *Deliver* | p, [DATA, s, m] $\rangle$ **do**
12: $ack[m] := ack[m] \cup \{p\}$
13: **if** $(s, m) \notin pending$ **then**
14: $pending := pending \cup \{(s, m)\}$
15: **trigger** $\langle$ *beb*, *Broadcast* | [DATA, s, m] $\rangle$
16: **function** SHOULDDELIVER(m)
17: **return** $|ack[m]| > \frac{N}{2} \wedge m \notin delivered$
18: **upon event** $\langle \exists_{(s,m) \in pending}$SHOULDDELIVER(m) $\rangle$ **do**
19: $delivered := delivered \cup \{m\}$
20: **trigger** $\langle$ *urb*, *Deliver* | s, m $\rangle$