Algorithm – Causal-Order Broadcast

Algorithm 1 No-Waiting Causal Broadcast

Implements:

CausalOrderReliableBroadcast, **instance** *crb*.

Uses:

ReliableBroadcast, **instance** *rb*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $\quad$ *delivered* $:= \emptyset$
3: $\quad$ *past* $:= []$
4: **upon event** $\langle$ *crb*, *Broadcast* | m $\rangle$ **do**
5: $\quad$ **trigger** $\langle$ *rb*, *Broadcast* | [DATA, *past*, m] $\rangle$
6: $\quad$ *past* $:= (self, m) :: past$ $\quad \triangleright$ List cons instead of append for brevity.
7: **upon event** $\langle$ *rb*, *Deliver* | p, [DATA, *mpast*, m] $\rangle$ **do**
8: $\quad$ **if** $m \notin delivered$ **then**
9: $\quad\quad$ DELIVERDEPS(*mpast*)
10: $\quad\quad$ **trigger** $\langle$ *crb*, *Deliver* | p, m $\rangle$
11: $\quad\quad$ *delivered* $:=$ *delivered* $\cup \{m\}$
12: $\quad\quad$ **if** $(p, m) \notin past$ **then**
13: $\quad\quad\quad$ *past* $:= (p, m) :: past$
14: **function** DELIVERDEPS($(p, m) :: rest$)
15: $\quad$ **if** $m \notin delivered$ **then**
16: $\quad\quad$ DELIVERDEPS($rest$)
17: $\quad\quad$ **trigger** $\langle$ *crb*, *Deliver* | p, m $\rangle$
18: $\quad\quad$ *delivered* $:=$ *delivered* $\cup \{m\}$
19: $\quad\quad$ **if** $(p, m) \notin past$ **then**
20: $\quad\quad\quad$ *past* $:= (p, m) :: past$

Algorithm 2 Broadcast with Sequence Number

Implements:

FIFOReliableBroadcast, **instance** *frb*.

Uses:

ReliableBroadcast, **instance** *rb*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $lsn := 0$
3: $pending := \emptyset$
4: $\forall_{p \in \Pi}\ next[p] := 1$
5: **upon event** $\langle$ *frb*, *Broadcast* | m $\rangle$ **do**
6: $lsn := lsn + 1$
7: **trigger** $\langle$ *rb*, *Broadcast* | [DATA, *self*, m, lsn] $\rangle$
8: **upon event** $\langle$ *rb*, *Deliver* | p, [DATA, s, m, sn] $\rangle$ **do**
9: $pending := pending \cup \{(s, m, sn)\}$
10: **while** $\exists_{(s,m',sn') \in pending}\ sn' = next[s]$ **do**
11: $next[s] := next[s] + 1$
12: $pending := pending \setminus \{(s, m', sn')\}$
13: **trigger** $\langle$ *frb*, *Deliver* | s, m' $\rangle$

Algorithm 3 No-Waiting Causal Broadcast with FIFO

Implements:

CausalOrderReliableBroadcast, **instance** *crb*.

Uses:

FIFO-ReliableBroadcast, **instance** *frb*.

```
1:  upon event ⟨ Init ⟩ do
2:      delivered := ∅
3:      l := []
4:  upon event ⟨ crb, Broadcast | m ⟩ do
5:      trigger ⟨ frb, Broadcast | [DATA, (self, m) :: l] ⟩
6:      l := []
7:  upon event ⟨ frb, Deliver | p, [DATA, l_m] ⟩ do
8:      deliverDeps(l_m)
9:  function deliverDeps((p, m) :: rest)
10:     if m ∉ delivered then
11:         deliverDeps(rest)
12:         trigger ⟨ crb, Deliver | p, m ⟩
13:         delivered := delivered ∪ {m}
14:         if (p, m) ∉ l then
15:             l := (p, m) :: l
```

Algorithm 4 Waiting Causal Broadcast

Implements:

CausalOrderReliableBroadcast, **instance** *crb*.

Uses:

ReliableBroadcast, **instance** *rb*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $\quad \forall_{p \in \Pi}\ V[p] := 0$
3: $\quad lsn := 0$
4: $\quad pending := \emptyset$
5: **upon event** $\langle$ *crb*, *Broadcast* | m $\rangle$ **do**
6: $\quad W := V$
7: $\quad W[self] := lsn$
8: $\quad lsn := lsn + 1$
9: $\quad$ **trigger** $\langle$ *rb*, *Broadcast* | $[\text{DATA}, W, m]$ $\rangle$
10: **upon event** $\langle$ *rb*, *Deliver* | $p, [\text{DATA}, W, m]$ $\rangle$ **do**
11: $\quad pending := pending \cup \{(p, W, m)\}$
12: $\quad$ **while** $\exists_{(p', W', m') \in pending}\ W' \leq V$ **do**
13: $\quad\quad pending := pending \setminus \{(p', W', m')\}$
14: $\quad\quad V[p'] := V[p'] + 1$
15: $\quad\quad$ **trigger** $\langle$ *crb*, *Deliver* | p', m' $\rangle$