

Algorithm – Causal-Order Broadcast

Algorithm 1 No-Waiting Causal Broadcast

Implements:

CausalOrderReliableBroadcast, **instance** *crb*.

Uses:

ReliableBroadcast, **instance** *rb*.

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

Algorithm 2 Broadcast with Sequence Number

Implements:FIFOReliableBroadcast, **instance** *frb*.**Uses:**ReliableBroadcast, **instance** *rb*.

```
1: upon event  $\langle \textit{Init} \rangle$  do
2:    $lsn := 0$ 
3:    $pending := \emptyset$ 
4:    $\forall_{p \in \Pi} next[p] := 1$ 
5: upon event  $\langle frb, Broadcast \mid m \rangle$  do
6:    $lsn := lsn + 1$ 
7:   trigger  $\langle rb, Broadcast \mid [DATA, self, m, lsn] \rangle$ 
8: upon event  $\langle rb, Deliver \mid p, [DATA, s, m, sn] \rangle$  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  $\langle frb, Deliver \mid s, m' \rangle$ 
```

Algorithm 3 No-Waiting Causal Broadcast with FIFO

Implements:

CausalOrderReliableBroadcast, **instance** *crb*.

Uses:

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

```
1: upon event  $\langle \textit{Init} \rangle$  do
2:    $\textit{delivered} := \emptyset$ 
3:    $l := []$ 
4: upon event  $\langle \textit{crb}, \textit{Broadcast} \mid m \rangle$  do
5:   trigger  $\langle \textit{frb}, \textit{Broadcast} \mid [\textit{DATA}, (\textit{self}, m) :: l] \rangle$ 
6:    $l := []$ 
7: upon event  $\langle \textit{frb}, \textit{Deliver} \mid p, [\textit{DATA}, l_m] \rangle$  do
8:   DELIVERDEPS( $l_m$ )
9: function DELIVERDEPS( $(p, m) :: \textit{rest}$ )
10:  if  $m \notin \textit{delivered}$  then
11:    DELIVERDEPS( $\textit{rest}$ )
12:    trigger  $\langle \textit{crb}, \textit{Deliver} \mid p, m \rangle$ 
13:     $\textit{delivered} := \textit{delivered} \cup \{m\}$ 
14:    if  $(p, m) \notin l$  then
15:       $l := (p, m) :: l$ 
```

Algorithm 4 Waiting Causal Broadcast

Implements:CausalOrderReliableBroadcast, **instance** *crb*.**Uses:**ReliableBroadcast, **instance** *rb*.

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