

Algorithm – Atomic Register

Algorithm 1 Read-Impose Write-Majority

Implements:

$(1, N)$ -AtomicRegister, **instance** *onar*.

Uses:

BestEffortBroadcast, **instance** *beb*.

PerfectPointToPointLinks, **instance** *pp2p*.

```

1: upon event  $\langle \textit{Init} \rangle$  do
2:    $(ts, val) := (0, \perp)$ 
3:    $wts := 0$ 
4:    $acks := 0$ 
5:    $rid := 0$ 
6:    $\forall p \in \Pi \text{ readlist}[p] := \perp$ 
7:    $readval := \perp$ 
8:    $reading := \text{FALSE}$ 
9: upon event  $\langle \textit{onar}, \textit{Read} \rangle$  do
10:   $rid := rid + 1$ 
11:   $acks := 0$ 
12:   $\forall p \in \Pi \text{ readlist}[p] := \perp$ 
13:   $reading := \text{TRUE}$ 
14:  trigger  $\langle \textit{beb}, \textit{Broadcast} \mid [\text{READ}, rid] \rangle$ 
15: upon event  $\langle \textit{beb}, \textit{Deliver} \mid p, [\text{READ}, r] \rangle$  do
16:   trigger  $\langle \textit{pp2p}, \textit{Send} \mid p, [\text{VALUE}, r, ts, val] \rangle$ 
17: upon event  $\langle \textit{pp2p}, \textit{Deliver} \mid p, [\text{VALUE}, r, ts', v'] \rangle$  do
18:   if  $r = rid$  then
19:      $readlist[p] := (ts', v')$ 
20:     if  $|\text{readlist}| > \frac{N}{2}$  then  $\triangleright$  Where  $N = |\Pi|$ .
21:        $(maxts, readval) := \text{HIGHEST}(\text{readlist})$ 
22:        $\forall q \in \Pi \text{ readlist}[q] := \perp$ 
23:       trigger  $\langle \textit{beb}, \textit{Broadcast} \mid [\text{WRITE}, rid, maxts, readval] \rangle$ 

```

```

24: upon event  $\langle \text{onar}, \text{Write} \mid v \rangle$  do
25:    $\text{rid} := \text{rid} + 1$ 
26:    $\text{wts} := \text{wts} + 1$ 
27:    $\text{acks} := 0$ 
28:   trigger  $\langle \text{beb}, \text{Broadcast} \mid [\text{WRITE}, \text{rid}, \text{wts}, v] \rangle$ 
29: upon event  $\langle \text{beb}, \text{Deliver} \mid p, [\text{WRITE}, r, \text{ts}', v'] \rangle$  do
30:   if  $\text{ts}' > \text{ts}$  then
31:      $(\text{ts}, \text{val}) := (\text{ts}', v')$ 
32:   trigger  $\langle \text{pp2p}, \text{Send} \mid p, [\text{ACK}, r] \rangle$ 
33: upon event  $\langle \text{pp2p}, \text{Deliver} \mid p, [\text{ACK}, r] \rangle$  do
34:   if  $r = \text{rid}$  then
35:      $\text{acks} := \text{acks} + 1$ 
36:     if  $\text{acks} > \frac{N}{2}$  then
37:        $\text{acks} := 0$ 
38:     if  $\text{reading}$  then
39:        $\text{reading} := \text{FALSE}$ 
40:       trigger  $\langle \text{onar}, \text{ReadReturn} \mid \text{readval} \rangle$ 
41:     else
42:       trigger  $\langle \text{onar}, \text{WriteReturn} \mid \rangle$ 

```

Algorithm 2 Read-Impose Write-Consult-Majority

Implements:

(N, N) -AtomicRegister, **instance** *nnar*.

Uses:

BestEffortBroadcast, **instance** *beb*.

PerfectPointToPointLinks, **instance** *pp2p*.

```
1: upon event  $\langle \text{Init} \rangle$  do
2:    $(ts, wr, val) := (0, 0, \perp)$ 
3:    $acks := 0$ 
4:    $writeval := \perp$ 
5:    $rid := 0$ 
6:    $\forall p \in \Pi \text{ readlist}[p] := \perp$ 
7:    $readval := \perp$ 
8:    $reading := \text{FALSE}$ 
9: upon event  $\langle \text{nnar}, \text{Read} \rangle$  do
10:   $rid := rid + 1$ 
11:   $acks := 0$ 
12:   $\forall p \in \Pi \text{ readlist}[p] := \perp$ 
13:   $reading := \text{TRUE}$ 
14:  trigger  $\langle \text{beb}, \text{Broadcast} \mid [\text{READ}, rid] \rangle$ 
15: upon event  $\langle \text{beb}, \text{Deliver} \mid p, [\text{READ}, r] \rangle$  do
16:   trigger  $\langle \text{pp2p}, \text{Send} \mid p, [\text{VALUE}, r, ts, wr, val] \rangle$ 
17: upon event  $\langle \text{pp2p}, \text{Deliver} \mid p, [\text{VALUE}, r, ts', wr', v'] \rangle$  do
18:   if  $r = rid$  then
19:      $readlist[p] := (ts', wr', v')$ 
20:     if  $|readlist| > \frac{N}{2}$  then  $\triangleright$  Where  $N = |\Pi|$ .
21:        $(maxts, rr, readval) := \text{HIGHEST}(readlist)$ 
22:        $\forall q \in \Pi \text{ readlist}[q] := \perp$ 
23:       if  $reading$  then
24:          $bcastval := readval$ 
25:       else
26:          $rr := \text{RANK}(self)$ 
27:          $maxts := maxts + 1$ 
28:          $bcastval := writeval$ 
29:       trigger  $\langle \text{beb}, \text{Broadcast} \mid [\text{WRITE}, rid, maxts, rr, bcastval] \rangle$ 
```

```

30: upon event  $\langle nnar, Write \mid v \rangle$  do
31:    $rid := rid + 1$ 
32:    $writeval := v$ 
33:    $acks := 0$ 
34:    $\forall_{p \in \Pi} readlist[p] := \perp$ 
35:   trigger  $\langle beb, Broadcast \mid [READ, rid] \rangle$ 
36: upon event  $\langle beb, Deliver \mid p, [WRITE, r, ts', wr', v'] \rangle$  do
37:   if  $(ts', wr') > (ts, wr)$  then  $\triangleright$  Tuple comparison.
38:      $(ts, wr, val) := (ts', wr', v')$ 
39:   trigger  $\langle pp2p, Send \mid p, [ACK, r] \rangle$ 
40: upon event  $\langle pp2p, Deliver \mid p, [ACK, r] \rangle$  do
41:   if  $r = rid$  then
42:      $acks := acks + 1$ 
43:     if  $acks > \frac{N}{2}$  then
44:        $acks := 0$ 
45:     if  $reading$  then
46:        $reading := \text{FALSE}$ 
47:       trigger  $\langle nnar, ReadReturn \mid readval \rangle$ 
48:     else
49:       trigger  $\langle nnar, WriteReturn \mid \rangle$ 

```
