Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Description

Each phase has two rounds, round 1 and round 2.

R[p,t] is a multi-set of received values in round t at phase p.

MAJ(R) = the number of occurrence of the majority non-? value in a multi-set R.

MajV(R) = the majority non-? value in a multi-set R.

Line 33 makes sure that every node flips the coin same number of times.

```
(i, p, 1, *) == round-1 messages == State messages
(i, p, 2, *) == round-2 messages == Vote messages
```

The verbose version of Weak-MVC and its helper function

Algorithm 4 IDK-BC: Code for node i

```
Local Variables: /* These variables can be accessed and modified by any thread at i. */
                                                                                    \trianglerightinput at node i, \{0,1\}
    v[t,1]
                                                          \trianglerighta vector of local state at round 1, \{0,1\}
    v[t,2]
                                                       \trianglerighta vector of local state at round 2, \{0,1,?\}
                                                                                              ⊳phase, integer
    p
/* All messages tagged with seq */
                                                         28:
                                                                else if MAJ(R[p,1]) \ge \lfloor \frac{n}{2} \rfloor + 1 then
When IDK-BC(v, seq) is invoked:
                                                         29:
                                                                   v[p,2] \leftarrow \text{MAJV}(R[p,1])
/* Initial Phase */
                                                         30:
                                                                \mathbf{else}
 1: Send (i, 0, 1, v) to all
                                                                   v[p,2] \leftarrow ?
                                                        31:
 2: wait until |R[0,1]| \ge n - f
 3: if MAJ(R[0,1]) \ge \lfloor \frac{n}{2} \rfloor + f + 1 then
                                                                /* Round 2 */
       Send (i, MAJV(R[0,1]), DEC) to all
                                                        32:
                                                                Send (i, p, 2, v[p, 2]) to all
                                                                randBit \leftarrow CommonCoinFlip()
       Return MAJV(R[0, 1])
                                                         33:
 5:
 6: else if MAJ(R[0,1]) \ge \lfloor \frac{n}{2} \rfloor + 1 then
                                                                wait until |R[p,2]| \geq n-f
                                                        34:
       v[0,2] \leftarrow 1
                                                                if MAJ(R[p,2]) \ge f+1 then
 7:
                                                        35:
                                                                   m \leftarrow \text{FindReturnValue}(p, 2)
 8: else
                                                         36:
 9:
       v[0,2] \leftarrow ?
                                                        37:
                                                                   Send (i, m, DEC) to all
An optimization for increasing 1's,
                                                         38:
                                                                   Return m
                                                                else if MAJ(R[p, 2]) \ge 1 then
 which might not be necessary.
                                                         39:
10: Send (i, 0, 2, v[0, 2]) to all
                                                                   v[p+1,1] \leftarrow \mathrm{MAJV}(R[p,2])
                                                         40:
11: wait until |R[0,2]| \ge n - f
                                                         41:
                                                                else
12: if MAJ(R[0,2]) \ge f+1 then
                                                         42:
                                                                   v[p+1,1] \leftarrow randBit
       m \leftarrow \text{FindReturnValue}(0, 2)
                                                                p \leftarrow p + 1
                                                                                    ⊳Proceed to next phase
13:
                                                        43:
       Send (i, m, DEC) to all
14:
                                                        /* Helper procedure */
       Return m
15:
16: else if MAJ(R[0,2]) \ge 1 then
                                                        Procedure FindReturnValue(p, r)
       v[1,1] \leftarrow \text{MAJV}(R[0,2])
                                                         44: if MAJV(R[p,r]) = 1 then
17:
                                                                if MAJ(R[0,1]) \ge \lfloor \frac{n}{2} \rfloor + 1 then
18: else
                                                        45:
                                                                  m \leftarrow \mathrm{MajV}(R[0,1])
19: v[1,1] \leftarrow 0
                                                        47:
                                                                else
/* Same as BO-BC except for the return step*/48:
                                                                   m \leftarrow \text{proposal from } j \text{ who has } 1 \text{ in } R[0,2]
20: p \leftarrow 1
                                                                Return m
                                                         49:
21: while TRUE do
                                                        50: else
       /* Round 1 */
                                                        51:
                                                                {f Return} \perp
       Send (i, p, 1, v[p, 1]) to all
22:
       wait until |R[p,1]| \ge n - f
                                                         /* Event handler: executing in background */
23:
       if MAJ(R[p,1]) \ge \lfloor \frac{n}{2} \rfloor + f + 1 then
                                                        Upon receiving (j, p, t, b) from node j:
24:
          m \leftarrow \text{FindReturnValue}(p, 1)
                                                        52: Add b to R[p,t]
25:
          Send (i, m, DEC) to all
                                                        Upon receiving (j, b, DEC) from node j:
26:
          Return m
                                                         53: Return b
27:
```