```
MODULE AJupiter
Specification of the Jupiter protocol presented by Attiya et al.
EXTENDS JupiterInterface
VARIABLES
    cbuf,    cbuf[c]: buffer for locally generated operations at client c ∈ Client
    crec,    crec[c]: number of remote operations received by client c ∈ Client
                      since the last time a local operation was generated
    sbuf,    sbuf[c]: buffer for transformed remote operations w.r.t client c ∈ Client
    srec     srec[c]: number of locally generated operations by client c ∈ Client
                      since the last time a remote operation was transformed at the Server

vars ≜ ⟨intVars, cbuf, crec, sbuf, srec⟩

AJMsg ≜
    [c : Client, ack : Nat, op : Op ∪ {Nop}] ∪ messages sent to the Server from client c ∈ Client
    [ack : Nat, op : Op ∪ {Nop}] messages broadcast to Clients from the Server
TypeOK ≜
    ∧ TypeOKInt
    ∧ cbuf ∈ [Client → Seq(Op ∪ {Nop})]
    ∧ crec ∈ [Client → Nat]
    ∧ sbuf ∈ [Client → Seq(Op ∪ {Nop})]
    ∧ srec ∈ [Client → Nat]
Init ≜
    ∧ InitInt
    ∧ cbuf = [c ∈ Client ↦ ⟨⟩]
    ∧ crec = [c ∈ Client ↦ 0]
    ∧ sbuf = [c ∈ Client ↦ ⟨⟩]
    ∧ srec = [c ∈ Client ↦ 0]
ClientPerform(c, m) ≜
    LET cBuf ≜ cbuf[c]
        cShiftedBuf ≜ SubSeq(cBuf, m.ack + 1, Len(cBuf))
        xop ≜ XformOpOps(Xform, m.op, cShiftedBuf)
        xcBuf ≜ XformOpsOp(Xform, cShiftedBuf, m.op)
    IN  ∧ cbuf′ = [cbuf EXCEPT ![c] = xcBuf]
        ∧ crec′ = [crec EXCEPT ![c] = @ + 1]
        ∧ SetNewAop(c, xop)

ServerPerform(m) ≜
    LET c ≜ m.c
        cBuf ≜ sbuf[c]
        cShiftedBuf ≜ SubSeq(cBuf, m.ack + 1, Len(cBuf))
```

```
48            xop ≜ XformOpOps(Xform, m.op, cShiftedBuf)
49             xcBuf ≜ XformOpsOp(Xform, cShiftedBuf, m.op)
50      IN     ∧ srec′ = [cl ∈ Client ↦
51                          IF cl = c THEN srec[cl] + 1 ELSE 0]
52             ∧ sbuf′ = [cl ∈ Client ↦
53                          IF cl = c THEN xcBuf ELSE Append(sbuf[cl], xop)]
54             ∧ SetNewAop(Server, xop)
55             ∧ Comm!SSend(c, [cl ∈ Client ↦ [ack ↦ srec[cl], op ↦ xop]])
56 ├──────────────────────────────────────────────────────────────────────┤
57  DoOp(c, op) ≜
58        ∧ SetNewAop(c, op)
59        ∧ cbuf′ = [cbuf EXCEPT ![c] = Append(@, op)]
60        ∧ crec′ = [crec EXCEPT ![c] = 0]
61        ∧ Comm!CSend([c ↦ c, ack ↦ crec[c], op ↦ op])

63  Do(c) ≜
64        ∧ DoInt(DoOp, c)
65        ∧ UNCHANGED ⟨sbuf, srec⟩

67  Rev(c) ≜
68        ∧ RevInt(ClientPerform, c)
69        ∧ UNCHANGED ⟨sbuf, srec⟩

71  SRev ≜
72       ∧ SRevInt(ServerPerform)
73       ∧ UNCHANGED ⟨cbuf, crec⟩
74 ├──────────────────────────────────────────────────────────────────────┤
75  Next ≜
76       ∨ ∃ c ∈ Client : Do(c) ∨ Rev(c)
77       ∨ SRev

79  Fairness ≜
80      WF_vars(SRev ∨ ∃ c ∈ Client : Rev(c))

82  Spec ≜ Init ∧ □[Next]_vars  ∧ Fairness
83 ├──────────────────────────────────────────────────────────────────────┤
84  QC ≜  Quiescent Consistency
85       Comm!EmptyChannel ⇒ Cardinality(Range(state)) = 1

87  THEOREM Spec ⇒ □QC
88 └──────────────────────────────────────────────────────────────────────┘
    \* Modification History
    \* Last modified Wed Jan 02 21:37:02 CST 2019 by hengxin
    \* Created Satchins, Jun 23 17:14:18 CST 2018 by hengxin
```