```
──────────────────── MODULE AJupiter ────────────────────
Specification of the Jupiter protocol presented by Hagit Attiya and others.
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⟩

Msg ≜ [c : Client, ack : Int, op : Op ∪ {Nop}] ∪   messages sent to the Server from a client c ∈ Client
      [ack : Int, op : Op ∪ {Nop}]   messages broadcast to Clients from the Server
─────────────────────────────────────────────────────────
TypeOK ≜
    ∧  TypeOKInt
    ∧  Comm(Msg)!TypeOK
    ∧  cbuf ∈ [Client → Seq(Op ∪ {Nop})]
    ∧  crec ∈ [Client → Int]
    ∧  sbuf ∈ [Client → Seq(Op ∪ {Nop})]
    ∧  srec ∈ [Client → Int]
─────────────────────────────────────────────────────────
Init ≜
    ∧ InitInt
    ∧ Comm(Msg)!Init
    ∧ cbuf = [c ∈ Client ↦ ⟨⟩]
    ∧ crec = [c ∈ Client ↦ 0]
    ∧ sbuf = [c ∈ Client ↦ ⟨⟩]
    ∧ srec = [c ∈ Client ↦ 0]
─────────────────────────────────────────────────────────
DoOp(c, op) ≜
      ∧ state′ = [state EXCEPT ![c] = Apply(op, @)]
      ∧ cbuf′ = [cbuf EXCEPT ![c] = Append(@, op)]
      ∧ crec′ = [crec EXCEPT ![c] = 0]
      ∧ Comm(Msg)!CSend([c ↦ c, ack ↦ crec[c], op ↦ op])

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

Rev(c) ≜
     ∧ Comm(Msg)!CRev(c)
     ∧ crec′ = [crec EXCEPT ![c] = @ + 1]
    ∧ LET m ≜ Head(cincoming[c])
          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]
          ∧ state′ = [state EXCEPT ![c] = Apply(xop, @)]
    ∧ RevInt(c)
    ∧ UNCHANGED ⟨sbuf, srec⟩

SRev ≜
    ∧ Comm(Msg)!SRev
    ∧ LET m ≜ Head(sincoming)
          c ≜ m.c
          cBuf ≜ sbuf[c]
          cShiftedBuf ≜ SubSeq(cBuf, m.ack + 1, Len(cBuf))
          xop ≜ XformOpOps(Xform, m.op, cShiftedBuf)
          xcBuf ≜ XformOpsOp(Xform, cShiftedBuf, m.op)
      IN  ∧ srec′ = [cl ∈ Client ↦
                        IF cl = c THEN srec[cl] + 1 ELSE 0]
          ∧ sbuf′ = [cl ∈ Client ↦
                        IF cl = c THEN xcBuf ELSE Append(sbuf[cl], xop)]
          ∧ state′ = [state EXCEPT ![Server] = Apply(xop, @)]
          ∧ Comm(Msg)!SSend(c, [cl ∈ Client ↦ [ack ↦ srec[cl], op ↦ xop]])
    ∧ SRevInt
    ∧ UNCHANGED ⟨cbuf, crec⟩
--------------------------------------------------------------------------------
Next ≜
    ∨ ∃ c ∈ Client : Do(c) ∨ Rev(c)
    ∨ SRev

Fairness ≜   There is no requirement that the clients ever generate operations.
    WF_vars(SRev ∨ ∃ c ∈ Client : Rev(c))

Spec ≜ Init ∧ □[Next]_vars  ∧ Fairness
--------------------------------------------------------------------------------
QC ≜   Quiescent Consistency
    Comm(Msg)!EmptyChannel ⇒ Cardinality(Range(state)) = 1

THEOREM Spec ⇒ □QC
================================================================================
\* Modification History
\* Last modified Mon Dec 31 21:02:17 CST 2018 by hengxin
\* Created Satchins, Jun 23 17:14:18 CST 2018 by hengxin
```