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

  5 EXTENDS JupiterInterface

  6 ├─────────────────────────────────────────────────────────────────┤
  7 VARIABLES
  8     cbuf,        cbuf[c]: buffer for locally generated operations at client c ∈ Client
  9     crec,        crec[c]: number of remote operations received by client c ∈ Client
 10                          since the last time a local operation was generated
 11     sbuf,        sbuf[c]: buffer for transformed remote operations w.r.t client c ∈ Client
 12     srec         srec[c]: number of locally generated operations by client c ∈ Client
 13                          since the last time a remote operation was transformed at the Server
```

15 $vars \triangleq \langle intVars, cbuf, crec, sbuf, srec \rangle$

17 $Msg \triangleq [c : Client, ack : Int, op : Op \cup \{Nop\}] \cup$ messages sent to the Server from a client $c \in Client$
18 $\qquad [ack : Int, op : Op \cup \{Nop\}]$ messages broadcast to Clients from the Server

```
 19 ├─────────────────────────────────────────────────────────────────┤
```
20 $TypeOK \triangleq$
21 $\quad \land \quad TypeOKInt$
22 $\quad \land \quad Comm(Msg)!TypeOK$
23 $\quad \land \quad cbuf \in [Client \to Seq(Op \cup \{Nop\})]$
24 $\quad \land \quad crec \in [Client \to Int]$
25 $\quad \land \quad sbuf \in [Client \to Seq(Op \cup \{Nop\})]$
26 $\quad \land \quad srec \in [Client \to Int]$

```
 27 ├─────────────────────────────────────────────────────────────────┤
```
28 $Init \triangleq$
29 $\quad \land InitInt$
30 $\quad \land Comm(Msg)!Init$
31 $\quad \land cbuf = [c \in Client \mapsto \langle\rangle]$
32 $\quad \land crec = [c \in Client \mapsto 0]$
33 $\quad \land sbuf = [c \in Client \mapsto \langle\rangle]$
34 $\quad \land srec = [c \in Client \mapsto 0]$

```
 35 ├─────────────────────────────────────────────────────────────────┤
```
36 $DoOp(c, op) \triangleq$
37 $\quad \land SetNewAop(c, op)$
38 $\quad \land cbuf' = [cbuf \text{ EXCEPT } ![c] = Append(@, op)]$
39 $\quad \land crec' = [crec \text{ EXCEPT } ![c] = 0]$
40 $\quad \land Comm(Msg)!CSend([c \mapsto c, ack \mapsto crec[c], op \mapsto op])$

42 $Do(c) \triangleq$
43 $\quad \land DoInt(DoOp, c)$
44 $\quad \land \text{UNCHANGED } \langle sbuf, srec \rangle$

46 $Rev(c) \triangleq$
47 $\quad \land Comm(Msg)!CRev(c)$
48 $\quad \land crec' = [crec \text{ EXCEPT } ![c] = @ + 1]$

1

$$
\begin{array}{rl}
49 & \quad \wedge \text{LET } m \overset{\triangle}{=} Head(cincoming[c]) \\
50 & \qquad\qquad cBuf \overset{\triangle}{=} cbuf[c] \\
51 & \qquad\qquad cShiftedBuf \overset{\triangle}{=} SubSeq(cBuf, m.ack + 1, Len(cBuf)) \\
52 & \qquad\qquad xop \overset{\triangle}{=} XformOpOps(Xform, m.op, cShiftedBuf) \\
53 & \qquad\qquad xcBuf \overset{\triangle}{=} XformOpsOp(Xform, cShiftedBuf, m.op) \\
54 & \qquad \text{IN} \quad \wedge cbuf' = [cbuf \text{ EXCEPT } ![c] = xcBuf] \\
55 & \qquad\qquad\quad \wedge SetNewAop(c, xop) \\
56 & \quad \wedge RevInt(c) \\
57 & \quad \wedge \text{UNCHANGED } \langle sbuf, srec \rangle
\end{array}
$$

$$
\begin{array}{rl}
59 & SRev \overset{\triangle}{=} \\
60 & \quad \wedge Comm(Msg)!SRev \\
61 & \quad \wedge \text{LET } m \overset{\triangle}{=} Head(sincoming) \\
62 & \qquad\qquad c \overset{\triangle}{=} m.c \\
63 & \qquad\qquad cBuf \overset{\triangle}{=} sbuf[c] \\
64 & \qquad\qquad cShiftedBuf \overset{\triangle}{=} SubSeq(cBuf, m.ack + 1, Len(cBuf)) \\
65 & \qquad\qquad xop \overset{\triangle}{=} XformOpOps(Xform, m.op, cShiftedBuf) \\
66 & \qquad\qquad xcBuf \overset{\triangle}{=} XformOpsOp(Xform, cShiftedBuf, m.op) \\
67 & \qquad \text{IN} \quad \wedge srec' = [cl \in Client \mapsto \\
68 & \qquad\qquad\qquad\qquad\qquad \text{IF } cl = c \text{ THEN } srec[cl] + 1 \text{ ELSE } 0] \\
69 & \qquad\qquad\quad \wedge sbuf' = [cl \in Client \mapsto \\
70 & \qquad\qquad\qquad\qquad\qquad \text{IF } cl = c \text{ THEN } xcBuf \text{ ELSE } Append(sbuf[cl], xop)] \\
71 & \qquad\qquad\quad \wedge SetNewAop(Server, xop) \\
72 & \qquad\qquad\quad \wedge Comm(Msg)!SSend(c, [cl \in Client \mapsto [ack \mapsto srec[cl], op \mapsto xop]]) \\
73 & \quad \wedge SRevInt \\
74 & \quad \wedge \text{UNCHANGED } \langle cbuf, crec \rangle
\end{array}
$$

75 ├──

$$
\begin{array}{rl}
76 & Next \overset{\triangle}{=} \\
77 & \quad \vee \exists\, c \in Client : Do(c) \vee Rev(c) \\
78 & \quad \vee SRev
\end{array}
$$

$$
\begin{array}{rl}
80 & Fairness \overset{\triangle}{=} \quad \boxed{\text{There is no requirement that the clients ever generate operations.}} \\
81 & \quad \text{WF}_{vars}(SRev \vee \exists\, c \in Client : Rev(c))
\end{array}
$$

$$
\begin{array}{rl}
83 & Spec \overset{\triangle}{=} Init \wedge \square[Next]_{vars} \;\boxed{\wedge\ Fairness}
\end{array}
$$

84 ├──

$$
\begin{array}{rl}
85 & QC \overset{\triangle}{=} \quad \boxed{\text{Quiescent Consistency}} \\
86 & \quad Comm(Msg)!EmptyChannel \Rightarrow Cardinality(Range(state)) = 1
\end{array}
$$

88 THEOREM $Spec \Rightarrow \square QC$

89 └──