MODULE *P4RuntimeElection*

EXTENDS *Naturals*, *Sequences*, *Controller*, *Device*

VARIABLE *step*

A sequence of all variables

$vars \triangleq \langle mastershipVars,\ nodeVars,\ messageVars,\ streamVars,\ deviceVars \rangle$

The invariant asserts that the device will not allow a write from an older master if it has already accepted a write from a newer master. This is determined by comparing the *mastership* terms of accepted writes. For this invariant to hold, terms may only increase in the history of writes.

$$
\begin{array}{l}
TypeInvariant \triangleq \\
\quad \land \forall\, x \in 1\,..\,Len(history) : \\
\qquad \forall\, y \in x\,..\,Len(history) : \\
\qquad\quad \land\ history[x].term \leq history[y].term \\
\qquad\quad \land\ history[x].term = history[y].term \Rightarrow history[x].node = history[y].node
\end{array}
$$

$$
\begin{array}{l}
Init \triangleq \\
\quad \land\ term = 0 \\
\quad \land\ master = Nil \\
\quad \land\ backups = \langle\rangle \\
\quad \land\ events = [n \in Nodes \mapsto \langle\rangle] \\
\quad \land\ mastership = [n \in Nodes \mapsto [term \mapsto 0,\ master \mapsto Nil,\ backups \mapsto \langle\rangle]] \\
\quad \land\ streamId\ \ = 0 \\
\quad \land\ sentTerm = [n \in Nodes \mapsto 0] \\
\quad \land\ isMaster\ \ = [n \in Nodes \mapsto \text{FALSE}] \\
\quad \land\ requestStream\ \ = [n \in Nodes \mapsto [id \mapsto 0,\ state \mapsto Closed]] \\
\quad \land\ requests = [n\ \ \in Nodes \mapsto \langle\rangle] \\
\quad \land\ responseStream = [n \in Nodes \mapsto [id \mapsto 0,\ state \mapsto Closed]] \\
\quad \land\ responses = [n\ \in Nodes \mapsto \langle\rangle] \\
\quad \land\ election = [n \in Nodes \mapsto 0] \\
\quad \land\ state = Stopped \\
\quad \land\ history = \langle\rangle
\end{array}
$$

$$
\begin{array}{l}
Next \triangleq \\
\quad \lor \exists\, n \in Nodes : OpenStream(n) \\
\qquad \land \text{UNCHANGED } \langle deviceVars \rangle \\
\quad \lor \exists\, n \in Nodes : CloseStream(n) \\
\qquad \land \text{UNCHANGED } \langle deviceVars \rangle \\
\quad \lor \exists\, n \in Nodes : ConnectStream(n) \\
\qquad \land \text{UNCHANGED } \langle mastershipVars,\ nodeVars \rangle \\
\quad \lor \exists\, n \in Nodes : DisconnectStream(n) \\
\qquad \land \text{UNCHANGED } \langle mastershipVars,\ nodeVars \rangle \\
\quad \lor \exists\, n \in Nodes : JoinMastershipElection(n) \\
\qquad \land \text{UNCHANGED } \langle deviceVars \rangle \\
\quad \lor \exists\, n \in Nodes : LeaveMastershipElection(n)
\end{array}
$$

$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor \exists\, n \in Nodes : LearnMastership(n)$
$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor \exists\, n \in Nodes : SendMasterArbitrationUpdate(n)$
$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor \exists\, n \in Nodes : HandleMasterArbitrationUpdate(n)$
$\land$ UNCHANGED $\langle mastershipVars,\ nodeVars \rangle$
$\lor \exists\, n \in Nodes : ReceiveMasterArbitrationUpdate(n)$
$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor \exists\, n \in Nodes : SendWriteRequest(n)$
$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor \exists\, n \in Nodes : HandleWrite(n)$
$\land$ UNCHANGED $\langle mastershipVars,\ nodeVars \rangle$
$\lor \exists\, n \in Nodes : ReceiveWriteResponse(n)$
$\land$ UNCHANGED $\langle deviceVars \rangle$
$\lor Shutdown$
$\land$ UNCHANGED $\langle mastershipVars,\ nodeVars \rangle$
$\lor Startup$
$\land$ UNCHANGED $\langle mastershipVars,\ nodeVars \rangle$

$Spec \triangleq Init \land \Box[Next]_{vars}$

* Modification History
* Last modified *Wed Mar* 27 15:51:45 *PDT* 2019 by *jordanhalterman*
* Created *Thu Feb* 14 11:33:03 *PST* 2019 by *jordanhalterman*