MODULE *Mastership*

EXTENDS *Southbound*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

CONSTANT *TraceMastership*

A record of target masterships
VARIABLE *mastership*

LOCAL *InitState* $\triangleq$
$[nodes \mapsto node,$
$\ mastership \mapsto mastership]$

LOCAL *NextState* $\triangleq$
$[nodes \mapsto node',$
$\ mastership \mapsto mastership']$

LOCAL *Trace* $\triangleq$ INSTANCE *Trace* WITH
Module $\leftarrow$ "Mastership",
InitState $\leftarrow$ *InitState*,
NextState $\leftarrow$ *NextState*,
Enabled $\leftarrow$ *TraceMastership*

This section models *mastership* reconciliation.

$$
\begin{array}{l}
ReconcileMastership(n) \triangleq \\
\quad \land \lor \land node[n].connected \\
\qquad\quad \land mastership.master = Nil \\
\qquad\quad \land mastership' = [master \mapsto n,\ term \mapsto mastership.term + 1,\ conn \mapsto node[n].incarnation] \\
\qquad \lor \land \neg node[n].connected \\
\qquad\quad \land mastership.master = n \\
\qquad\quad \land mastership' = [mastership \text{ EXCEPT } !.master = Nil] \\
\quad \land \text{UNCHANGED } \langle node,\ target \rangle
\end{array}
$$

Formal specification, constraints, and theorems.

$$
\begin{array}{l}
InitMastership \triangleq \\
\quad \land mastership = [master \mapsto Nil,\ term \mapsto 0,\ conn \mapsto 0]
\end{array}
$$

$$NextMastership \triangleq$$
$$\quad \lor \exists\, n \in Nodes :$$
$$\qquad Trace!Step(ReconcileMastership(n),\ [node \mapsto n])$$

* Modification History
* Last modified Sun *Feb* 20 09:09:52 *PST* 2022 by *jordanhalterman*
* Created Sun *Feb* 20 03:13:26 *PST* 2022 by *jordanhalterman*