
MODULE *RIC*

LOCAL INSTANCE *Naturals*

LOCAL INSTANCE *Sequences*

LOCAL INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

An empty value

CONSTANT *Nil*

Node states

CONSTANT *Stopped, Started*

A set of *RIC* node identifiers

CONSTANT *RICNode*

ASSUME $\wedge \text{IsFiniteSet}(\text{RICNode})$
 $\wedge \forall n \in \text{RICNode} : n \in \text{STRING}$

A mapping of node states

VARIABLE *state*

Network state

VARIABLE *network*

A store of *E2* node states

VARIABLE *nodes*

$\text{vars} \triangleq \langle \text{state}, \text{network}, \text{nodes} \rangle$

LOCAL *E2AP* \triangleq INSTANCE *E2AP* WITH $\text{conns} \leftarrow \text{network}$

$\text{StartNode}(n) \triangleq$
 $\wedge \text{state}[n] = \text{Stopped}$
 $\wedge \text{state}' = [\text{nodes} \text{ EXCEPT } ![n] = \text{Started}]$
 $\wedge \text{UNCHANGED } \langle \text{network}, \text{nodes} \rangle$

$\text{StopNode}(n) \triangleq$
 $\wedge \text{state}[n] = \text{Started}$
 $\wedge \text{state}' = [\text{nodes} \text{ EXCEPT } ![n] = \text{Stopped}]$
 $\wedge \text{UNCHANGED } \langle \text{network}, \text{nodes} \rangle$

$$\begin{aligned}
& \text{HandleE2SetupRequest}(\text{node}, \text{conn}, \text{res}) \triangleq \\
& \quad \wedge \text{E2AP!Server}(\text{node})!\text{Reply!E2SetupResponse}(\text{conn}, [\text{foo} \mapsto \text{"bar"}, \text{bar} \mapsto \text{"baz"}]) \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleRICControlResponse}(\text{node}, \text{conn}, \text{res}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleRICSubscriptionResponse}(\text{node}, \text{conn}, \text{res}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleRICSubscriptionDeleteResponse}(\text{node}, \text{conn}, \text{res}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleRICIndication}(\text{node}, \text{conn}, \text{res}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleE2NodeConfigurationUpdate}(\text{node}, \text{conn}, \text{req}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{state} \rangle \\
& \text{HandleRequest}(\text{node}, \text{conn}) \triangleq \\
& \quad \wedge \vee \text{E2AP!Server}(\text{node})!\text{Handle!E2SetupRequest}(\text{conn}, \text{LAMBDA } c, m : \text{HandleE2SetupRequest}(\text{node}, \text{conn}, c, m)) \\
& \quad \vee \text{E2AP!Server}(\text{node})!\text{Handle!RICControlResponse}(\text{conn}, \text{LAMBDA } c, m : \text{HandleRICControlResponse}(\text{node}, \text{conn}, c, m)) \\
& \quad \vee \text{E2AP!Server}(\text{node})!\text{Handle!RICSubscriptionResponse}(\text{conn}, \text{LAMBDA } c, m : \text{HandleRICSubscriptionResponse}(\text{node}, \text{conn}, c, m)) \\
& \quad \vee \text{E2AP!Server}(\text{node})!\text{Handle!RICSubscriptionDeleteResponse}(\text{conn}, \text{LAMBDA } c, m : \text{HandleRICSubscriptionDeleteResponse}(\text{node}, \text{conn}, c, m)) \\
& \quad \vee \text{E2AP!Server}(\text{node})!\text{Handle!RICIndication}(\text{conn}, \text{LAMBDA } c, m : \text{HandleRICIndication}(\text{node}, \text{conn}, c, m)) \\
& \quad \vee \text{E2AP!Server}(\text{node})!\text{Handle!E2NodeConfigurationUpdate}(\text{conn}, \text{LAMBDA } c, m : \text{HandleE2NodeConfigurationUpdate}(\text{node}, \text{conn}, c, m)) \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{Init} \triangleq \\
& \quad \wedge \text{E2AP!Init} \\
& \quad \wedge \text{state} = [n \in \text{RICNode} \mapsto \text{Stopped}] \\
& \quad \wedge \text{nodes} = [n \in \{\} \mapsto [id \mapsto \text{Nil}]] \\
& \text{Next} \triangleq \\
& \quad \vee \exists \text{node} \in \text{RICNode} : \text{StartNode}(\text{node}) \\
& \quad \vee \exists \text{node} \in \text{RICNode} : \text{StopNode}(\text{node}) \\
& \quad \vee \exists \text{node} \in \text{RICNode} : \exists \text{conn} \in \text{E2AP!RIC}(\text{node})!\text{Connections} : \text{HandleRequest}(\text{node}, \text{conn})
\end{aligned}$$

\ * Modification History
\ * Last modified Tue Sep 21 08:37:34 PDT 2021 by jordanhalterman
\ * Created Tue Sep 21 02:14:49 PDT 2021 by jordanhalterman