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

A mapping of node states

VARIABLE *state*

Network state

VARIABLE *net*

A store of *E2* node states

VARIABLE *nodes*

$vars \triangleq \langle state, net, nodes \rangle$

LOCAL *E2AP* \triangleq INSTANCE *E2AP* WITH $conns \leftarrow net$

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

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

$HandleE2SetupRequest(node, conn, res) \triangleq$
 $\wedge E2AP!RIC(node)!Reply!E2SetupResponse(conn, [foo \mapsto \text{"bar"}, bar \mapsto \text{"baz"}])$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleRICControlResponse(node, conn, res) \triangleq$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleRICSubscriptionResponse(node, conn, res) \triangleq$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleRICSubscriptionDeleteResponse(node, conn, res) \triangleq$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleRICIndication(node, conn, res) \triangleq$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleE2NodeConfigurationUpdate(node, conn, req) \triangleq$
 $\wedge \text{UNCHANGED } \langle state \rangle$

$HandleRequest(node, conn) \triangleq$
 $\wedge \vee E2AP!RIC(node)!Handle!E2SetupRequest(conn, \text{LAMBDA } m : HandleE2SetupRequest(node, conn, m))$
 $\vee E2AP!RIC(node)!Handle!RICControlResponse(conn, \text{LAMBDA } m : HandleRICControlResponse(node, conn, m))$
 $\vee E2AP!RIC(node)!Handle!RICSubscriptionResponse(conn, \text{LAMBDA } m : HandleRICSubscriptionResponse(node, conn, m))$
 $\vee E2AP!RIC(node)!Handle!RICSubscriptionDeleteResponse(conn, \text{LAMBDA } m : HandleRICSubscriptionDeleteResponse(node, conn, m))$
 $\vee E2AP!RIC(node)!Handle!RICIndication(conn, \text{LAMBDA } m : HandleRICIndication(node, conn, m))$
 $\vee E2AP!RIC(node)!Handle!E2NodeConfigurationUpdate(conn, \text{LAMBDA } m : HandleE2NodeConfigurationUpdate(node, conn, m))$
 $\wedge \text{UNCHANGED } \langle nodes \rangle$

$Init \triangleq$
 $\wedge E2AP!Init$
 $\wedge state = [n \in RICNode \mapsto Stopped]$
 $\wedge nodes = [n \in \{\} \mapsto [id \mapsto Nil]]$

$Next \triangleq$
 $\vee \exists node \in RICNode : StartNode(node)$
 $\vee \exists node \in RICNode : StopNode(node)$
 $\vee \exists node \in RICNode : \exists conn \in E2AP!RIC(node)!Connections : HandleRequest(node, conn)$

\backslash * Modification History
 \backslash * Last modified Tue Sep 21 06:07:29 PDT 2021 by jordanhalterman
 \backslash * Created Tue Sep 21 02:14:49 PDT 2021 by jordanhalterman