MODULE *RANSim*

LOCAL INSTANCE *Naturals*

LOCAL INSTANCE *Sequences*

LOCAL INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

An empty value
CONSTANT *Nil*

Node states
CONSTANT *Stopped*, *Started*

Connection states
CONSTANT *Connecting*, *Connected*, *Configuring*, *Configured*

The set of *E*2 node identifiers
CONSTANT *E2Node*

ASSUME $\land$ *IsFiniteSet*(*E2Node*)
$\land \forall\, n \in E2Node : n \in$ STRING

A set of *RIC* node identifiers
CONSTANT *RICNode*

ASSUME $\land$ *IsFiniteSet*(*RICNode*)
$\land \forall\, n \in RICNode : n \in$ STRING

The state of the *E*2 node
VARIABLE *state*

The state of the network
VARIABLE *network*

The primary management connection
VARIABLE *mgmtConn*

The state of *E*2*AP* connections
VARIABLE *dataConn*

The set of outstanding transactions
VARIABLE *transactions*

Subscriptions
VARIABLE *subs*

$vars \triangleq \langle state,\ network,\ mgmtConn,\ dataConn,\ subs \rangle$

LOCAL $E2AP \triangleq$ INSTANCE $E2AP$ WITH $conns \leftarrow network$

$StartNode(e2Node) \triangleq$
$\quad \land\ state[e2Node] = Stopped$
$\quad \land\ state' = [state \text{ EXCEPT } ![e2Node] = Started]$
$\quad \land$ UNCHANGED $\langle network,\ mgmtConn,\ dataConn,\ subs \rangle$

$StopNode(e2Node) \triangleq$
$\quad \land\ state[e2Node] = Started$
$\quad \land\ state' = [state \text{ EXCEPT } ![e2Node] = Stopped]$
$\quad \land$ UNCHANGED $\langle network,\ mgmtConn,\ dataConn,\ subs \rangle$

$ReconcileConnection(e2NodeId,\ ricNodeId) \triangleq$
$\quad \land\ ricNodeId \in dataConn[e2NodeId]$
$\quad \land\ \lor\ \land\ dataConn[e2NodeId].state = Connecting$
$\qquad \land\ E2AP!Client(e2NodeId)!Connect(ricNodeId)$
$\qquad \land$ LET $newConnId \triangleq$ CHOOSE $i \in \{conn.id : conn \in network[e2NodeId]\} : i \notin \{conn.id : conn \in net$
$\qquad\quad$ IN
$\qquad\qquad \land\ dataConn' = [dataConn \text{ EXCEPT } ![e2NodeId] = dataConn[e2NodeId] \mathbin{@@} (ricNodeId :> [state$
$\qquad\qquad \land$ UNCHANGED $\langle transactions \rangle$
$\qquad\ \lor\ \land\ dataConn[e2NodeId].state \neq Connecting$
$\qquad\quad \land\ \lor\ \land \exists\, conn \in E2AP!Client(e2NodeId)!Connections :$
$\qquad\qquad\qquad \land\ conn.id = dataConn[e2NodeId].conn$
$\qquad\qquad\qquad \land\ \lor\ \land\ dataConn[e2NodeId].state = Connecting$
$\qquad\qquad\qquad\qquad \land\ dataConn' = [dataConn \text{ EXCEPT } ![e2NodeId] = [$
$\qquad\qquad\qquad\qquad\qquad\qquad dataConn[e2NodeId] \text{ EXCEPT } ![ricNodeId].state = Connected]]$
$\qquad\qquad\qquad\qquad \land$ UNCHANGED $\langle transactions \rangle$
$\qquad\qquad\qquad\ \lor\ \land\ dataConn[e2NodeId].state = Connected$
$\qquad\qquad\qquad\qquad \land\ Len(transactions[e2NodeId]) < 256$
$\qquad\qquad\qquad\qquad \land$ LET $txId \triangleq$ CHOOSE $i \in 0\,..\,255 : i \notin$ DOMAIN $transactions[e2NodeId]$
$\qquad\qquad\qquad\qquad\qquad\quad req \triangleq [txId \mapsto txId,\ e2NodeId \mapsto e2NodeId]$
$\qquad\qquad\qquad\qquad\quad$ IN
$\qquad\qquad\qquad\qquad\qquad \land\ E2AP!Client(e2NodeId)!Send!E2NodeConfigurationUpdate(conn,\ req)$
$\qquad\qquad\qquad\qquad\qquad \land\ transactions' = [transactions \text{ EXCEPT } ![e2NodeId] = transactions[e2NodeId] \mathbin{@@}$
$\qquad\qquad\qquad\qquad\qquad \land\ dataConn' = [dataConn \text{ EXCEPT } ![e2NodeId] = [$
$\qquad\qquad\qquad\qquad\qquad\qquad dataConn[e2NodeId] \text{ EXCEPT } ![ricNodeId].state = Configuring]]$
$\qquad\qquad\qquad\ \lor\ \land\ dataConn[e2NodeId].state = Configuring$
$\qquad\qquad\qquad\qquad \land\ E2AP!Client(e2NodeId)!Ready(conn)$
$\qquad\qquad\qquad\qquad \land$ LET $res \triangleq E2AP!Client(e2NodeId)!Read(conn)$
$\qquad\qquad\qquad\qquad\quad$ IN
$\qquad\qquad\qquad\qquad\qquad \land\ E2AP!Client(e2NodeId)!Receive!E2NodeConfigurationUpdateAcknowledge(cor$

$\wedge\ dataConn' = [dataConn \text{ EXCEPT } ![e2NodeId] = [$
$dataConn[e2NodeId] \text{ EXCEPT } ![ricNodeId].state = Configured]]$
$\wedge \text{ UNCHANGED } \langle transactions \rangle$
$\vee\ \wedge\ dataConn[e2NodeId].state = Configured$
$\wedge \text{ UNCHANGED } \langle dataConn \rangle$
$\vee\ \wedge \neg\exists\, conn \in E2AP!Client(e2NodeId)!Connections : conn.id = dataConn[e2NodeId].conn$
$\wedge\ dataConn' = [dataConn \text{ EXCEPT } ![e2NodeId] = [$
$dataConn[e2NodeId] \text{ EXCEPT } ![ricNodeId] = [state \mapsto Connecting,\ conn \mapsto N$
$\wedge \text{ UNCHANGED } \langle subs \rangle$

$Connect(e2NodeId,\ ricNodeId) \triangleq$
$\wedge\ E2AP!Client(e2NodeId)!Connect(ricNodeId)$
$\wedge \text{ UNCHANGED } \langle state,\ dataConn,\ transactions \rangle$

$Disconnect(e2NodeId,\ conn) \triangleq$
$\wedge\ E2AP!Client(e2NodeId)!Disconnect(conn)$
$\wedge \text{ UNCHANGED } \langle state,\ dataConn,\ transactions \rangle$

$E2Setup(e2NodeId,\ conn) \triangleq$
$\wedge \neg\exists\, c \in E2AP!Client(e2NodeId)!Connections : c.id = mgmtConn[e2NodeId].connId$
$\wedge\ Len(transactions[e2NodeId]) < 256$
$\wedge \text{ LET } txId \triangleq \text{ CHOOSE } i \in 0\,..\,255 : i \notin \text{DOMAIN } transactions$
$req \triangleq [txId \mapsto txId,\ e2NodeId \mapsto E2Node]$
IN
$\wedge\ transactions' = transactions \mathbin{@@} (txId :> req)$
$\wedge\ E2AP!Client(E2Node)!Send!E2SetupRequest(conn,\ req)$
$\wedge \text{ UNCHANGED } \langle mgmtConn,\ dataConn,\ subs \rangle$

$HandleE2SetupResponse(e2NodeId,\ conn,\ res) \triangleq$
$\wedge\ E2AP!Client(E2Node)!Receive!E2SetupResponse(conn,\ res)$
$\wedge\ \vee\ \wedge\ res.txId \in \text{DOMAIN } transactions[e2NodeId]$
$\wedge\ mgmtConn' = [mgmtConn \text{ EXCEPT } ![e2NodeId] = [connId \mapsto conn.id]]$
$\wedge\ transactions' = [transactions \text{ EXCEPT } ![e2NodeId] = [$
$t \in \text{DOMAIN } transactions[e2NodeId] \setminus \{res.txId\} \mapsto transactions[e2NodeId][t]]]$
$\vee\ \wedge\ res.txId \notin transactions[e2NodeId]$
$\wedge \text{ UNCHANGED } \langle mgmtConn,\ transactions \rangle$
$\wedge \text{ UNCHANGED } \langle dataConn,\ subs \rangle$

$HandleRICSubscriptionRequest(e2NodeId,\ conn,\ req) \triangleq$
$\wedge\ E2AP!Client(E2Node)!Receive!RICSubscriptionRequest(conn,\ req)$
$\wedge \text{ UNCHANGED } \langle dataConn,\ subs \rangle$

$HandleRICSubscriptionDeleteRequest(e2NodeId,\ conn,\ req) \triangleq$
$\wedge\ E2AP!Client(E2Node)!Receive!RICSubscriptionDeleteRequest(conn,\ req)$
$\wedge \text{ UNCHANGED } \langle dataConn,\ subs \rangle$

$HandleRICControlRequest(e2NodeId,\ conn,\ req) \triangleq$
$\land\ E2AP!Client(E2Node)!Receive!RICControlRequest(conn,\ req)$
$\land\ E2AP!Client(E2Node)!Reply!RICControlAcknowledge(conn,\ [foo \mapsto \text{“bar”},\ bar \mapsto \text{“baz”}])$
$\land$ UNCHANGED $\langle dataConn,\ subs \rangle$

$HandleE2ConnectionUpdate(e2NodeId,\ conn,\ req) \triangleq$
$\land\ E2AP!Client(E2Node)!Receive!E2ConnectionUpdate(conn,\ req)$
$\land$ LET $add \triangleq$ IF “add” $\in$ DOMAIN req THEN $req[\text{“add”}]$ ELSE $\{\}$
$update \triangleq$ IF “update” $\in$ DOMAIN req THEN $req[\text{“update”}]$ ELSE $\{\}$
$remove \triangleq$ IF “remove” $\in$ DOMAIN req THEN $req[\text{“remove”}]$ ELSE $\{\}$
IN
$\land\ dataConn' = [dataConn$ EXCEPT $![e2NodeId] = [$
$n \in (\text{DOMAIN}\ dataConn[e2NodeId] \cup add) \setminus remove \mapsto$
IF $n \notin update \land n \in$ DOMAIN $dataConn$ THEN
$dataConn[n]$
ELSE
$[state \mapsto Connecting,\ conn \mapsto Nil]]]$
$\land$ UNCHANGED $\langle subs \rangle$

$HandleE2NodeConfigurationUpdateAcknowledge(e2NodeId,\ conn,\ res) \triangleq$
$\land\ E2AP!Client(E2Node)!Receive!E2NodeConfigurationUpdateAcknowledge(conn,\ res)$
$\land\ res.txId \in transactions$
$\land\ dataConn[conn.dst].state = Configuring$
$\land\ transactions' = [t \in$ DOMAIN $transactions \setminus \{res.txId\} \mapsto transactions[t]]$
$\land\ dataConn' = [dataConn$ EXCEPT $![conn.dst].state = Configured]$
$\land$ UNCHANGED $\langle subs \rangle$

$HandleRequest(e2NodeId,\ conn) \triangleq$
$\land\ \lor E2AP!Client(E2Node)!Handle!RICSubscriptionRequest(conn,$ LAMBDA $c,\ m : HandleRICSubscriptio$
$\lor E2AP!Client(E2Node)!Handle!RICSubscriptionDeleteRequest(conn,$ LAMBDA $c,\ m : HandleRICSubs$
$\lor E2AP!Client(E2Node)!Handle!RICControlRequest(conn,$ LAMBDA $c,\ m : HandleRICControlRequest$
$\lor E2AP!Client(E2Node)!Handle!E2ConnectionUpdate(conn,$ LAMBDA $c,\ m : HandleE2ConnectionUp$
$\lor E2AP!Client(E2Node)!Handle!E2NodeConfigurationUpdateAcknowledge(conn,$ LAMBDA $c,\ m : Hand$
$\land$ UNCHANGED $\langle state \rangle$

$Init \triangleq$
$\land\ E2AP!Init$
$\land\ state = [n \in E2Node \mapsto Stopped]$
$\land\ mgmtConn = [n \in E2Node \mapsto [connId \mapsto Nil]]$
$\land\ dataConn\ \ = [n \in E2Node \mapsto [c \in \{\} \mapsto [connId \mapsto Nil]]]$
$\land\ transactions = [n \in E2Node \mapsto [t \in \{\} \mapsto [id\ \ \mapsto Nil]]]$
$\land\ subs = [n \in E2Node \mapsto [i \in \{\} \mapsto [id\ \ \mapsto Nil]]]$

$Next \triangleq$
$\lor \exists\ e2NodeId \in E2Node :$

$$
\begin{array}{l}
\quad StartNode(e2NodeId) \\
\lor \exists\, e2NodeId \in E2Node : \\
\quad StopNode(e2NodeId) \\
\lor \exists\, e2NodeId \in E2Node,\ ricNodeId \in RICNode : \\
\quad Connect(e2NodeId,\ ricNodeId) \\
\lor \exists\, e2NodeId \in E2Node,\ ricNodeId \in RICNode : \\
\quad Disconnect(e2NodeId,\ ricNodeId) \\
\lor \exists\, e2NodeId \in E2Node : \\
\quad \exists\, conn \in E2AP!Client(e2NodeId)!Connections : \\
\qquad E2Setup(e2NodeId,\ conn) \\
\lor \exists\, e2NodeId \in E2Node : \\
\quad \exists\, conn \in E2AP!Client(e2NodeId)!Connections : \\
\qquad HandleRequest(e2NodeId,\ conn)
\end{array}
$$

* Modification History
* Last modified *Tue Sep* 21 15:04:44 *PDT* 2021 by *jordanhalterman*
* Created *Tue Sep* 21 13:27:29 *PDT* 2021 by *jordanhalterman*