MODULE $E2Node$

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 $E2$ node identifier
CONSTANT $E2Node$

A set of RIC node identifiers
CONSTANT RIC

ASSUME $\land\ IsFiniteSet(RIC)$
$\land\ \forall\, n \in RIC : n \in \text{STRING}$

The state of the $E2$ node
VARIABLE $state$

The state of the network
VARIABLE $network$

The primary management connection
VARIABLE $mgmtConn$

The state of $E2AP$ 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 $conns \leftarrow network$

$$StartNode \triangleq$$
$$\land\ state = Stopped$$
$$\land\ state' = Started$$
$$\land\ \text{UNCHANGED}\ \langle network,\ mgmtConn,\ dataConn,\ subs\rangle$$

$$StopNode \triangleq$$
$$\land\ state = Started$$
$$\land\ state' = Stopped$$
$$\land\ \text{UNCHANGED}\ \langle network,\ mgmtConn,\ dataConn,\ subs\rangle$$

$$ConnectManagement(node) \triangleq$$
$$\land\ \neg\exists\, conn \in E2AP!Client(E2Node)!Connections : conn.dst = node$$
$$\land\ E2AP!Client(E2Node)!Connect(node)$$
$$\land\ \text{UNCHANGED}\ \langle state,\ mgmtConn,\ dataConn,\ subs\rangle$$

$$DisconnectManagement(conn) \triangleq$$
$$\land\ E2AP!Client(E2Node)!Disconnect(conn)$$
$$\land\ \text{UNCHANGED}\ \langle state,\ mgmtConn,\ dataConn,\ subs\rangle$$

$$ConnectDataConn(node) \triangleq$$
$$\land\ dataConn[node].state = Connecting$$
$$\land\ E2AP!Client(E2Node)!Connect(node)$$
$$\land\ dataConn' = [dataConn\ \text{EXCEPT}\ ![node].state = Connected]$$
$$\land\ \text{UNCHANGED}\ \langle state,\ mgmtConn,\ dataConn,\ subs\rangle$$

$$ConfigureDataConn(node) \triangleq$$
$$\land\ dataConn[node].state = Connected$$
$$\land\ \exists\, conn \in E2AP!Client(E2Node)!Connections : conn.dst = node$$
$$\land\ \text{LET}\ conn \triangleq \text{CHOOSE}\ c \in E2AP!Client(E2Node)!Connections : c.dst = node$$
$$\qquad txId \triangleq \text{CHOOSE}\ i \in 0\,..\,255 : i \notin \text{DOMAIN}\ transactions$$
$$\qquad req \triangleq [txId \mapsto txId,\ e2NodeId \mapsto E2Node]$$
$$\quad \text{IN}$$
$$\qquad \land\ transactions' = transactions\ @@\ (txId :> req)$$
$$\qquad \land\ dataConn' = [dataConn\ \text{EXCEPT}\ ![node].state = Configuring]$$
$$\qquad \land\ E2AP!Client(E2Node)!Send!E2NodeConfigurationUpdate(conn,\ req)$$
$$\land\ \text{UNCHANGED}\ \langle state,\ mgmtConn,\ subs\rangle$$

$$E2Setup(conn) \triangleq$$
$$\land\ Len(transactions) < 256$$
$$\land\ \text{LET}\ txId \triangleq \text{CHOOSE}\ i \in 0\,..\,255 : i \notin \text{DOMAIN}\ transactions$$
$$\qquad req \triangleq [txId \mapsto txId,\ e2NodeId \mapsto E2Node]$$
$$\quad \text{IN}$$
$$\qquad \land\ transactions' = transactions\ @@\ (txId :> req)$$

$\land\ E2AP!Client(E2Node)!Send!E2SetupRequest(conn,\ req)$
$\land$ UNCHANGED $\langle mgmtConn,\ dataConn,\ subs\rangle$

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

$HandleRICSubscriptionRequest(conn,\ req)\ \triangleq$
$\land\ E2AP!Client(E2Node)!Receive!RICSubscriptionRequest(conn,\ req)$
$\land$ UNCHANGED $\langle dataConn,\ subs\rangle$

$HandleRICSubscriptionDeleteRequest(conn,\ req)\ \triangleq$
$\land\ E2AP!Client(E2Node)!Receive!RICSubscriptionDeleteRequest(conn,\ req)$
$\land$ UNCHANGED $\langle dataConn,\ subs\rangle$

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

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

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

$HandleRequest(c)\ \triangleq$

$\land \lor E2AP!Client(E2Node)!Handle!RICSubscriptionRequest(c,\ HandleRICSubscriptionRequest)$
$\lor E2AP!Client(E2Node)!Handle!RICSubscriptionDeleteRequest(c,\ HandleRICSubscriptionDeleteRequ$
$\lor E2AP!Client(E2Node)!Handle!RICControlRequest(c,\ HandleRICControlRequest)$
$\lor E2AP!Client(E2Node)!Handle!E2ConnectionUpdate(c,\ HandleE2ConnectionUpdate)$
$\lor E2AP!Client(E2Node)!Handle!E2NodeConfigurationUpdateAcknowledge(c,\ HandleE2NodeConfigur$
$\land$ UNCHANGED $\langle state \rangle$

$Init \triangleq$
$\land E2AP!Init$
$\land state = Stopped$
$\land mgmtConn = [connId \mapsto Nil]$
$\land dataConn \ = [c \in \{\} \mapsto [connId \mapsto Nil]]$
$\land txId = 0$
$\land subs = [i \in \{\} \mapsto [id \mapsto Nil]]$

$Next \triangleq$
$\lor StartNode$
$\lor StopNode$
$\lor \exists\, node \in RIC : E2AP!Client(E2Node)!Connect(node)$
$\lor \exists\, conn \in E2AP!Client(E2Node)!Connections : E2AP!Client(E2Node)!Disconnect(conn)$
$\lor \exists\, conn \in E2AP!Client(E2Node)!Connections : SendE2SetupRequest(conn)$
$\lor \exists\, conn \in E2AP!Client(E2Node)!Connections : HandleRequest(conn)$

---

* Modification History
* Last modified *Tue Sep* 21 10:40:55 *PDT* 2021 by *jordanhalterman*
* Created *Tue Sep* 21 02:14:57 *PDT* 2021 by *jordanhalterman*