

---

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

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*

---

$StartNode \triangleq$   
 $\wedge state = Stopped$   
 $\wedge state' = Started$   
 $\wedge UNCHANGED \langle network, mgmtConn, dataConn, subs \rangle$

$StopNode \triangleq$   
 $\wedge state = Started$   
 $\wedge state' = Stopped$   
 $\wedge UNCHANGED \langle network, mgmtConn, dataConn, subs \rangle$

---

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

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

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

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

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

$$\begin{aligned}
& \wedge E2AP!Client(E2Node)!Send!E2SetupRequest(conn, req) \\
& \wedge \text{UNCHANGED } \langle mgmtConn, dataConn, subs \rangle \\
\\
HandleE2SetupResponse(conn, res) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!E2SetupResponse(conn, res) \\
& \wedge \vee \wedge res.txId \in transactions \\
& \quad \wedge mgmtConn' = conn \\
& \quad \wedge transactions' = [t \in \text{DOMAIN } transactions \setminus \{res.txId\} \mapsto transactions[t]] \\
& \vee \wedge res.txId \notin transactions \\
& \quad \wedge \text{UNCHANGED } \langle transactions \rangle \\
& \wedge \text{UNCHANGED } \langle dataConn, subs \rangle \\
\\
HandleRICSubscriptionRequest(conn, req) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!RICSubscriptionRequest(conn, req) \\
& \wedge \text{UNCHANGED } \langle dataConn, subs \rangle \\
\\
HandleRICSubscriptionDeleteRequest(conn, req) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!RICSubscriptionDeleteRequest(conn, req) \\
& \wedge \text{UNCHANGED } \langle dataConn, subs \rangle \\
\\
HandleRICControlRequest(conn, req) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!RICControlRequest(conn, req) \\
& \wedge E2AP!Client(E2Node)!Reply!RICControlAcknowledge(conn, [foo \mapsto "bar", bar \mapsto "baz"]) \\
& \wedge \text{UNCHANGED } \langle dataConn, subs \rangle \\
\\
HandleE2ConnectionUpdate(conn, req) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!E2ConnectionUpdate(conn, req) \\
& \wedge \text{LET } add \triangleq \text{IF "add"} \in \text{DOMAIN } req \text{ THEN } req["add"] \text{ ELSE } \{\} \\
& \quad update \triangleq \text{IF "update"} \in \text{DOMAIN } req \text{ THEN } req["update"] \text{ ELSE } \{\} \\
& \quad remove \triangleq \text{IF "remove"} \in \text{DOMAIN } req \text{ THEN } req["remove"] \text{ ELSE } \{\} \\
& \text{IN} \\
& \quad \wedge dataConn' = [n \in (\text{DOMAIN } dataConn \cup add) \cap remove \mapsto \\
& \quad \quad \text{IF } n \in \text{DOMAIN } dataConn \text{ THEN} \\
& \quad \quad \quad dataConn[n] \\
& \quad \quad \text{ELSE} \\
& \quad \quad \quad [state \mapsto Connecting]] \\
& \wedge \text{UNCHANGED } \langle subs \rangle \\
\\
HandleE2NodeConfigurationUpdateAcknowledge(conn, res) & \triangleq \\
& \wedge E2AP!Client(E2Node)!Receive!E2NodeConfigurationUpdateAcknowledge(conn, res) \\
& \wedge res.txId \in transactions \\
& \wedge dataConn[conn.dst].state = Configuring \\
& \wedge transactions' = [t \in \text{DOMAIN } transactions \setminus \{res.txId\} \mapsto transactions[t]] \\
& \wedge dataConn' = [dataConn \text{ EXCEPT } ![conn.dst].state = Configured] \\
& \wedge \text{UNCHANGED } \langle subs \rangle \\
\\
HandleRequest(c) & \triangleq
\end{aligned}$$

$$\begin{aligned}
& \wedge \vee E2AP!Client(E2Node)!Handle!RICSubscriptionRequest(c, HandleRICSubscriptionRequest) \\
& \vee E2AP!Client(E2Node)!Handle!RICSubscriptionDeleteRequest(c, HandleRICSubscriptionDeleteRequest) \\
& \vee E2AP!Client(E2Node)!Handle!RICControlRequest(c, HandleRICControlRequest) \\
& \vee E2AP!Client(E2Node)!Handle!E2ConnectionUpdate(c, HandleE2ConnectionUpdate) \\
& \vee E2AP!Client(E2Node)!Handle!E2NodeConfigurationUpdateAcknowledge(c, HandleE2NodeConfigurationUpdateAcknowledge) \\
& \wedge \text{UNCHANGED } \langle state \rangle
\end{aligned}$$


---

*Init*  $\triangleq$

$$\begin{aligned}
& \wedge E2AP!Init \\
& \wedge state = Stopped \\
& \wedge mgmtConn = [connId \mapsto Nil] \\
& \wedge dataConn = [c \in \{\}] \mapsto [connId \mapsto Nil] \\
& \wedge transactions = [t \in \{\}] \mapsto [id \mapsto Nil] \\
& \wedge subs = [i \in \{\}] \mapsto [id \mapsto Nil]
\end{aligned}$$

*Next*  $\triangleq$

$$\begin{aligned}
& \vee StartNode \\
& \vee StopNode \\
& \vee \exists node \in RIC : E2AP!Client(E2Node)!Connect(node) \\
& \vee \exists conn \in E2AP!Client(E2Node)!Connections : E2AP!Client(E2Node)!Disconnect(conn) \\
& \vee \exists conn \in E2AP!Client(E2Node)!Connections : E2Setup(conn) \\
& \vee \exists conn \in E2AP!Client(E2Node)!Connections : HandleRequest(conn)
\end{aligned}$$


---

\ \* Modification History  
\ \* Last modified Tue Sep 21 13:36:09 PDT 2021 by jordanhalterman  
\ \* Created Mon Sep 13 15:18:28 PDT 2021 by jordanhalterman