```
┌──────────────────────── 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  ∧ IsFiniteSet(RICNode)
        ∧ ∀ n ∈ RICNode : n ∈ STRING

 A mapping of node states
VARIABLE state

 Network state
VARIABLE network

 A store of E2 node states
VARIABLE nodes

vars ≜ ⟨state, network, nodes⟩

LOCAL E2AP ≜ INSTANCE E2AP WITH conns ← network

├────────────────────────────────────────────────────────────┤

StartNode(n) ≜
   ∧ state[n] = Stopped
   ∧ state′ = [nodes EXCEPT ![n] = Started]
   ∧ UNCHANGED ⟨network, nodes⟩

StopNode(n) ≜
   ∧ state[n] = Started
   ∧ state′ = [nodes EXCEPT ![n] = Stopped]
   ∧ UNCHANGED ⟨network, nodes⟩

├────────────────────────────────────────────────────────────┤
```

```
HandleE2SetupRequest(node, conn, req) ≜
    ∧ E2AP!Server(node)!Receive!E2SetupRequest(conn, req)
    ∧ ∨ ∧ req.globalE2NodeId ∉ DOMAIN nodes
          ∧ nodes′ = nodes @@ (req.globalE2NodeId :> [globalE2NodeId ↦ req.globalE2NodeId,
                                                          serviceModels   ↦ req.serviceModels])
      ∨ ∧ req.globalE2NodeId ∈ DOMAIN nodes
          ∧ nodes′ = [nodes EXCEPT ![req.globalE2NodeId] = [
                        nodes[req.globalE2NodeId] EXCEPT !.serviceModels = req.serviceModels]]
    ∧ E2AP!Server(node)!Reply!E2SetupResponse(conn, [transactionId ↦ req.transactionId])
    ∧ UNCHANGED ⟨state⟩

HandleRICControlResponse(node, conn, res) ≜
    ∧ E2AP!Server(node)!Receive!RICControlResponse(conn, res)
    ∧ UNCHANGED ⟨state⟩

HandleRICSubscriptionResponse(node, conn, res) ≜
    ∧ E2AP!Server(node)!Receive!RICSubscriptionResponse(conn, res)
    ∧ UNCHANGED ⟨state⟩

HandleRICSubscriptionDeleteResponse(node, conn, res) ≜
    ∧ E2AP!Server(node)!Receive!RICSubscriptionDeleteResponse(conn, res)
    ∧ UNCHANGED ⟨state⟩

HandleRICIndication(node, conn, res) ≜
    ∧ E2AP!Server(node)!Receive!RICIndication(conn, res)
    ∧ UNCHANGED ⟨state⟩

HandleE2NodeConfigurationUpdate(node, conn, req) ≜
    ∧ E2AP!Server(node)!Receive!E2NodeConfigurationUpdate(conn, req)
    ∧ UNCHANGED ⟨state⟩

HandleRequest(node, conn) ≜
    ∧ ∨ E2AP!Server(node)!Handle!E2SetupRequest(conn, LAMBDA c, m : HandleE2SetupRequest(node, co
      ∨ E2AP!Server(node)!Handle!RICControlResponse(conn, LAMBDA c, m : HandleRICControlResponse
      ∨ E2AP!Server(node)!Handle!RICSubscriptionResponse(conn, LAMBDA c, m : HandleRICSubscription
      ∨ E2AP!Server(node)!Handle!RICSubscriptionDeleteResponse(conn, LAMBDA c, m : HandleRICSubsc
      ∨ E2AP!Server(node)!Handle!RICIndication(conn, LAMBDA c, m : HandleRICIndication(node, conn,
      ∨ E2AP!Server(node)!Handle!E2NodeConfigurationUpdate(conn, LAMBDA c, m : HandleE2NodeConfi
    ∧ UNCHANGED ⟨nodes⟩
```

```
Init ≜
    ∧ E2AP!Init
    ∧ state  = [n ∈ RICNode ↦ Stopped]
    ∧ nodes = [n ∈ {} ↦ [id ↦ Nil]]

Next ≜
```

$\lor \exists\, node \in RICNode : StartNode(node)$

$\lor \exists\, node \in RICNode : StopNode(node)$

$\lor \exists\, node \in RICNode : \exists\, conn \in E2AP!Server(node)!Connections : HandleRequest(node,\ conn)$

* Modification History
* Last modified *Tue Sep* 21 09:33:19 *PDT* 2021 by *jordanhalterman*
* Created *Tue Sep* 21 02:14:49 *PDT* 2021 by *jordanhalterman*