
MODULE *E2T*

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 *E2T* node identifiers

CONSTANT *E2TNodes*

ASSUME $\wedge IsFiniteSet(E2TNodes)$
 $\wedge \forall n \in E2TNodes : n \in \text{STRING}$

A set of *E2* node identifiers

CONSTANT *E2Nodes*

ASSUME $\wedge IsFiniteSet(E2Nodes)$
 $\wedge \forall n \in E2Nodes : n \in \text{STRING}$

A mapping of node states

VARIABLE *nodes*

A global store of mastership for each *E2* node

VARIABLE *masterships*

A global store of connections for each *E2* node

VARIABLE *conns*

A store of streams for each node

VARIABLE *streams*

A global store of channel states

VARIABLE *chans*

A global store of subscription states

VARIABLE *subs*

$vars \triangleq \langle nodes, masterships, conns, streams, chans, subs \rangle$

$$\begin{aligned}
& \text{StartNode}(n) \triangleq \\
& \quad \wedge \text{nodes}[n] = \text{Stopped} \\
& \quad \wedge \text{nodes}' = [\text{nodes} \text{ EXCEPT } ![n] = \text{Started}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{masterships}, \text{conns}, \text{streams}, \text{chans}, \text{subs} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{StopNode}(n) \triangleq \\
& \quad \wedge \text{nodes}[n] = \text{Started} \\
& \quad \wedge \text{nodes}' = [\text{nodes} \text{ EXCEPT } ![n] = \text{Stopped}] \\
& \quad \wedge \text{streams}' = [\text{streams} \text{ EXCEPT } ![n] = [id \in \{\} \mapsto [id \mapsto \text{Nil}]]] \\
& \quad \wedge \text{UNCHANGED } \langle \text{masterships}, \text{conns}, \text{chans}, \text{subs} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{HandleSubscribeRequest}(n, c, r) \triangleq \\
& \quad \wedge \vee \wedge r.\text{sub}.id \notin \text{streams}[n] \\
& \quad \quad \wedge \text{streams}' = [\text{streams} \text{ EXCEPT } ![n] = \text{streams}[n] @@ (r.\text{sub}.id :> [id \mapsto r.\text{sub}.id])] \\
& \quad \vee \wedge r.\text{sub}.id \in \text{streams}[n] \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{streams} \rangle \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes}, \text{chans}, \text{subs} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{ReconcileMastership}(n, e) \triangleq \\
& \quad \wedge \text{masterships}[e].\text{master} \notin \text{DOMAIN } \text{conns}[e] \\
& \quad \wedge \exists c \in \text{DOMAIN } \text{conns}[e] : c \neq \text{masterships}[e].\text{master} \\
& \quad \wedge \text{masterships}' = [\text{masterships} \text{ EXCEPT } ![e] = [\\
& \quad \quad \text{term} \mapsto \text{masterships}[e].\text{term} + 1, \\
& \quad \quad \text{conn} \mapsto \text{CHOOSE } c \in \text{DOMAIN } \text{conns}[e] : c \neq \text{masterships}[e].\text{master}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes}, \text{subs} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{ReconcileStream}(n, s) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes}, \text{subs} \rangle
\end{aligned}$$

ReconcileChannel reconciles a channel's state

$$\begin{aligned}
& \text{ReconcileChannel}(n, c) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes}, \text{streams} \rangle
\end{aligned}$$

ReconcileSubscription reconciles a subscription's state

$$\begin{aligned}
& \text{ReconcileSubscription}(n, s) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{nodes}, \text{streams}, \text{chans} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{Init} \triangleq \\
& \quad \wedge \text{nodes} = [n \in E2TNodes \mapsto \text{Stopped}] \\
& \quad \wedge \text{masterships} = [e \in E2Nodes \mapsto [\text{master} \mapsto \text{Nil}, \text{term} \mapsto 0]] \\
& \quad \wedge \text{conns} = [e \in E2Nodes \mapsto [c \in \{\} \mapsto [id \mapsto c, e2node \mapsto \text{Nil}, e2t \mapsto \text{Nil}]]] \\
& \quad \wedge \text{streams} = [n \in E2TNodes \mapsto [x \in \{\} \mapsto [id \mapsto x]]]
\end{aligned}$$

$$\wedge \text{chans} = [x \in \{\} \mapsto [id \mapsto x]]$$

$$\wedge \text{subs} = [x \in \{\} \mapsto [id \mapsto x]]$$

$Next \triangleq$

- $\vee \exists n \in E2TNodes :$
 - $StartNode(n)$
- $\vee \exists n \in E2TNodes :$
 - $StopNode(n)$
- $\vee \exists n \in E2TNodes, e \in E2Nodes :$
 - $ReconcileMastership(n, e)$
- $\vee \exists n \in E2TNodes :$
 - $\exists s \in streams[n] :$
 - $ReconcileStream(n, s)$
- $\vee \exists n \in E2TNodes, c \in chans :$
 - $ReconcileChannel(n, c)$
- $\vee \exists n \in E2TNodes, s \in subs :$
 - $ReconcileSubscription(n, s)$

* Modification History
* Last modified Mon Sep 13 16:35:22 PDT 2021 by jordanhalterman
* Created Mon Sep 13 03:23:39 PDT 2021 by jordanhalterman