MODULE *SCTP*

LOCAL INSTANCE *Naturals*

LOCAL INSTANCE *Sequences*

LOCAL INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

CONSTANT *Nil*

LOCAL $Min(s) \triangleq$ CHOOSE $x \in s : \forall\, y \in s : x \geq y$

LOCAL $Max(s) \triangleq$ CHOOSE $x \in s : \forall\, y \in s : x \leq y$

VARIABLE *conns*

$vars \triangleq \langle conns \rangle$

MODULE *Client*

CONSTANT *ID*

$Connect(tgt) \triangleq$
$\land\ tgt \in$ DOMAIN $conns$
$\land$ LET $maxId \triangleq Max(\{conns[tgt][i].connId : i \in conns[tgt]\})$
$connId \triangleq Min(\{i \in 1\,..\,(maxId + 1) : i \notin$ DOMAIN $conns[tgt]\})$
$conn \triangleq [id \mapsto connId,$
$src \mapsto ID,$
$tgt \mapsto tgt,$
$req \mapsto \langle\rangle,$
$res \mapsto \langle\rangle]$
IN $conns' = [conns$ EXCEPT $![tgt] = conns[tgt] \mathbin{@@} (connId :> conn)]$

$Disconnect(conn) \triangleq$
$conns' = [conns$ EXCEPT $![conn.tgt] = [x \in$ DOMAIN $conns[conn.tgt] \setminus \{conn.id\} \mapsto conns[conn.tgt][x]]]$

$Send(conn,\ msg) \triangleq$
$conns' = [conns$ EXCEPT $![conn.tgt] = [$
$conns[conn.tgt]$ EXCEPT $![conn.id] = [$
$conns[conn.tgt][conn.id]$ EXCEPT $!.req = Append(conns[conn.tgt][conn.id].req,\ msg)]]]$

$Receive(conn) \triangleq$
$conns' = [conns$ EXCEPT $![conn.tgt] = [$
$conns[conn.tgt]$ EXCEPT $![conn.id] = [$
$conns[conn.tgt][conn.id]$ EXCEPT $!.res = SubSeq(conns[conn.tgt][conn.id].res,\ 2,\ Len(co$

$Reply(conn,\ msg) \triangleq$
$conns' = [conns' \text{ EXCEPT } ![conn.tgt] = [$
$conns'[conn.tgt] \text{ EXCEPT } ![conn.id] = [$
$conns'[conn.tgt][conn.id] \text{ EXCEPT } !.req = Append(conns'[conn.tgt][conn.id].req,\ msg)]]]$

$Connections \triangleq \{conn \in \text{UNION } \{\{conns[s][c] : c \in \text{DOMAIN } s\} : s \in conns\} : conn.src = ID\}$

$Connected(connId) \triangleq \exists\, s \in conns : \exists\, c \in s : c.id = connId$

$Ready(conn) \triangleq Len(conn.res) > 0$

$Read(conn) \triangleq conn.res[1]$

$Client(ID) \triangleq \text{INSTANCE } Client$

MODULE *Server*

CONSTANT ID

$Start \triangleq$
$\land\ ID \notin \text{DOMAIN } conns$
$\land\ conns' = conns \mathbin{@@} (ID :> [connId \in \{\} \mapsto [connId \mapsto connId]])$

$Stop \triangleq$
$\land\ ID \in \text{DOMAIN } conns$
$\land\ conns' = [c \in \{c \in \text{DOMAIN } conns : c \neq ID\} \mapsto conns[c]]$

$Send(conn,\ msg) \triangleq$
$\land\ Assert(conn.tgt = ID,\ \text{“Send on invalid connection”})$
$\land\ conns' = [conns \text{ EXCEPT } ![conn.tgt] = [$
$conns[conn.tgt] \text{ EXCEPT } ![conn.id] = [$
$conns[conn.tgt][conn.id] \text{ EXCEPT } !.res = Append(conns[conn.tgt][conn.id].res,\ msg)]]$

$Receive(conn) \triangleq$
$\land\ Assert(conn.tgt = ID,\ \text{“Receive on invalid connection”})$
$\land\ conns' = [conns \text{ EXCEPT } ![conn.tgt] = [$
$conns[conn.tgt] \text{ EXCEPT } ![conn.id] = [$
$conns[conn.tgt][conn.id] \text{ EXCEPT } !.res = SubSeq(conns[conn.tgt][conn.id].req,\ 2,\ Len($

$Reply(conn,\ msg) \triangleq$
$\land\ Assert(conn.tgt = ID,\ \text{“Reply on invalid connection”})$
$\land\ conns' = [conns' \text{ EXCEPT } ![conn.tgt] = [$
$conns'[conn.tgt] \text{ EXCEPT } ![conn.id] = [$
$conns'[conn.tgt][conn.id] \text{ EXCEPT } !.req = Append(conns'[conn.tgt][conn.id].res,\ msg)]]]$

$Connections \triangleq \{conn \in \text{UNION } \{\{conns[s][c] : c \in \text{DOMAIN } s\} : s \in conns\} : conn.tgt = ID\}$

$Connected(connId) \triangleq \exists\, s \in conns : \exists\, c \in s : c.id = connId$

$Ready(conn) \triangleq Len(conn.req) > 0$

$Read(conn) \triangleq conn.req[1]$

$Server(ID) \triangleq \text{INSTANCE } Server$

$Init \triangleq$
$\land conns = [id \in \{\} \mapsto [$
$connId \in \{\} \mapsto [connId \mapsto connId,$
$src \mapsto Nil,$
$tgt \mapsto Nil,$
$req \mapsto \langle\rangle,$
$res \mapsto \langle\rangle]]]$

$Next \triangleq$
$\lor \text{UNCHANGED } \langle conns \rangle$

* Modification History
* Last modified *Tue Sep* 21 14:35:07 *PDT* 2021 by *jordanhalterman*
* Created *Mon Sep* 13 12:21:16 *PDT* 2021 by *jordanhalterman*