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*

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

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

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

$Receive(conn) \triangleq$
$\quad conns' = [conns$ EXCEPT $![conn.tgt] = [$
$\qquad conns[conn.tgt]$ EXCEPT $![conn.id] = [$

```
                    conns[conn.tgt][conn.id] EXCEPT !.res =
                    SubSeq(conns[conn.tgt][conn.id].res, 2, Len(conns[conn.tgt][conn.id].res))]]]
  Reply(conn, msg) ≜
    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)]]]

  Connections ≜ {conn ∈ UNION {{conns[s][c] : c ∈ DOMAIN s} : s ∈ DOMAIN conns} : conn.src = ID}

  Connected(connId) ≜ ∃ s ∈ conns : ∃ c ∈ s : c.id = connId

  Ready(conn) ≜ Len(conn.res) > 0

  Read(conn) ≜ conn.res[1]
```

```
Client(ID) ≜ INSTANCE Client
```

MODULE *Server*

```
  CONSTANT ID

  Start ≜
    ∧ ID ∉ DOMAIN conns
    ∧ conns' = conns @@ (ID :> [connId ∈ {} ↦ [connId ↦ connId]])

  Stop ≜
    ∧ ID ∈ DOMAIN conns
    ∧ conns' = [c ∈ {c ∈ DOMAIN conns : c ≠ ID} ↦ conns[c]]

  Send(conn, msg) ≜
    ∧ Assert(conn.tgt = ID, "Send on invalid connection")
    ∧ conns' = [conns EXCEPT ![conn.tgt] = [
                  conns[conn.tgt] EXCEPT ![conn.id] = [
                    conns[conn.tgt][conn.id] EXCEPT !.res =
                    Append(conns[conn.tgt][conn.id].res, msg)]]]

  Receive(conn) ≜
    ∧ Assert(conn.tgt = ID, "Receive on invalid connection")
    ∧ conns' = [conns EXCEPT ![conn.tgt] = [
                  conns[conn.tgt] EXCEPT ![conn.id] = [
                    conns[conn.tgt][conn.id] EXCEPT !.res =
                    SubSeq(conns[conn.tgt][conn.id].req, 2, Len(conns[conn.tgt][conn.id].req))]]]

  Reply(conn, msg) ≜
    ∧ Assert(conn.tgt = ID, "Reply on invalid connection")
```

$$
\begin{array}{l}
\quad \wedge\ conns' = [conns' \text{ EXCEPT } ![conn.tgt] = [\\
\qquad\qquad conns'[conn.tgt] \text{ EXCEPT } ![conn.id] = [\\
\qquad\qquad\quad conns'[conn.tgt][conn.id] \text{ EXCEPT } !.req = \\
\qquad\qquad\quad Append(conns'[conn.tgt][conn.id].res,\ msg)]]] \\
\\
\quad Connections \triangleq \{conn \in \text{UNION } \{\{conns[s][c] : c \in \text{DOMAIN } s\} : s \in \text{DOMAIN } conns\} : conn.tgt = ID\} \\
\\
\quad Connected(connId) \triangleq \exists\, s \in conns : \exists\, c \in s : c.id = connId \\
\\
\quad Ready(conn) \triangleq Len(conn.req) > 0 \\
\\
\quad Read(conn) \triangleq conn.req[1]
\end{array}
$$

$$
\begin{array}{l}
Server(ID) \triangleq \text{INSTANCE } Server \\
\\
Init \triangleq \\
\quad \wedge\ conns = [id \in \{\} \mapsto [\\
\qquad\qquad connId \in \{\} \mapsto [connId \mapsto connId, \\
\qquad\qquad\qquad\qquad\qquad\quad src \mapsto Nil, \\
\qquad\qquad\qquad\qquad\qquad\quad tgt \mapsto Nil, \\
\qquad\qquad\qquad\qquad\qquad\quad req \mapsto \langle\rangle, \\
\qquad\qquad\qquad\qquad\qquad\quad res \mapsto \langle\rangle]]] \\
\\
Next \triangleq \\
\quad \vee \text{UNCHANGED } \langle conns \rangle
\end{array}
$$

* Modification History
* Last modified *Wed Sep* 22 15:48:13 *PDT* 2021 by *adibrastegarnia*
* Last modified *Tue Sep* 21 14:35:07 *PDT* 2021 by *jordanhalterman*
* Created *Mon Sep* 13 12:21:16 *PDT* 2021 by *jordanhalterman*