MODULE *CSComm*

Specification of communication in a Client-Server system model.

EXTENDS *Integers*, *Naturals*, *OpOperators*

CONSTANTS
    Client, the set of clients
    Server, the (unique) server
    Op

VARIABLES
    cincoming, *cincoming*[*c*]: incoming channel at the client $c \in Client$
    sincoming incoming channel at the *Server*

$vars \triangleq \langle cincoming, sincoming \rangle$

Messages between the *Server* and the Clients. There are two kinds of messages according to their destinations. *TODO*: Abstraction from the concrete representation of messages.

$Msg \triangleq [c : Client, ack : Int, op : Op \cup \{Nop\}] \cup$ messages sent to the *Server* from a client $c \in Client$
$\quad [ack : Int, op : Op \cup \{Nop\}]$ messages broadcast to Clients from the *Server*

$TypeOK \triangleq$
$\quad \land \quad cincoming \in [Client \to Seq(Msg)]$
$\quad \land \quad sincoming \in Seq(Msg)$

The initial predicate.

$Init \triangleq$
$\quad \land cincoming = [c \in Client \mapsto \langle\rangle]$
$\quad \land sincoming = \langle\rangle$

A client sends a message *msg* to the *Server*.

$CSend(msg) \triangleq$
$\quad \land sincoming' = Append(sincoming, msg)$
$\quad \land$ UNCHANGED *cincoming*

A client receives a message from the *Server*.

$CRev(c) \triangleq$
$\quad \land cincoming[c] \neq \langle\rangle$ there are messages to handle with
$\quad \land cincoming' = [cincoming$ EXCEPT $![c] = Tail(@)]$ consume a message
$\quad \land$ UNCHANGED *sincoming*

SRev and *SSend* below will be used together in one subaction. Therefore, there are no UNCHANGED sub-formulas in their definitions.

The *Server* receives a message from some clinet $c \in Client$.

$SRev \triangleq$
$\quad \land sincoming \neq \langle\rangle$ there are messages for the *Server* to handle with

```
61      ∧ sincoming′ = Tail(sincoming)  consume a message
        The Server broadcasts messages to the Clients other than c ∈ Client. The "ack" parts of the
        messages [ack: Int, op: Op] broadcast are determined by the parameter "acks".
68  SSend(c, acks, xop) ≜
69      ∧ cincoming′ = [cl ∈ Client ↦
70                          IF cl = c
71                           THEN cincoming[cl]
72                           ELSE Append(cincoming[cl], [ack ↦ acks[cl], op ↦ xop])]
73 ├──────────────────────────────────────────────────────────────────────────┤
        Properties of communication.
77  EmptyChannel ≜ Init
78 └──────────────────────────────────────────────────────────────────────────┘
    \* Modification History
    \* Last modified Sun Sep 02 10:52:57 CST 2018 by hengxin
    \* Created Sun Jun 24 10:25:34 CST 2018 by hengxin
```