MODULE *HopProtocol*

EXTENDS *Integers*, *Naturals*, *TLC*, *Sequences*, *FiniteSets*

VARIABLES
l1Chain,
l2Chain,
pendingTransfers,
commitThreshold,
bondedWithdrawals,
roots

$Pick(S) \triangleq \text{CHOOSE } s \in S : \text{TRUE}$

RECURSIVE $SetReduce(\_, \_, \_)$
$SetReduce(Op(\_, \_), S, value) \triangleq$
IF $S = \{\}$ THEN $value$
ELSE LET $s \triangleq Pick(S)$
IN IF $Op(s, value) = Op(value, s)$
THEN $SetReduce(Op, S \setminus \{s\}, Op(s, value))$
ELSE $Assert(\text{FALSE}, \text{"Err"})$

$SumSeq(S) \triangleq$
LET $seq \triangleq S$
$Sum[i \in 1 \,..\, Len(seq)] \triangleq \text{IF } i = 1 \text{ THEN } seq[i] \text{ ELSE } seq[i] + Sum[i-1]$
IN IF $seq = \langle\rangle$ THEN 0 ELSE $Sum[Len(seq)]$

RECURSIVE $SeqFromSet(\_)$
$SeqFromSet(S) \triangleq$
IF $S = \{\}$ THEN $\langle\rangle$
ELSE LET $x \triangleq \text{CHOOSE } x \in S : \text{TRUE}$
IN $\langle x \rangle \circ SeqFromSet(S \setminus \{x\})$

$Hash(v) \triangleq \text{CHOOSE } n \in \{Int\} : \text{TRUE}$

ASSUME $(Hash(\langle 1 \rangle) = Hash(\langle 1 \rangle))$
ASSUME $(Hash(\langle \{1, 2, 3\} \rangle) = Hash(\langle \{2, 1, 3, 1\} \rangle))$

$TypeOK \triangleq \text{TRUE}$

$Init \triangleq$
$\land\ l1Chain = 1$
$\land\ l2Chain = 2 \,..\, 3$
$\land\ pendingTransfers = [c \in 2 \,..\, 3 \mapsto \langle\rangle]$
$\land\ commitThreshold = 2$
$\land\ roots = [c \in 1 \,..\, 3 \mapsto \langle\rangle]$

$\land\ bondedWithdrawals = [c \in 1\,..\,3 \mapsto \{\}]$

$SendTransfer(c)\ \triangleq$
$\land\ Len(pendingTransfers[c]) < 3$
$\land\ pendingTransfers' = [pendingTransfers\ \text{EXCEPT}\ ![c] = pendingTransfers[c] \circ \langle[$
$target \mapsto RandomElement(2\,..\,3),\ amount \mapsto RandomElement(1\,..\,2),\ id \mapsto RandomElement(1\,..\,2)$
$]\rangle]$
$\land\ Print(\langle \text{“send”} \rangle,\ \text{TRUE})$
$\land\ \text{UNCHANGED}\ \langle l1Chain,\ l2Chain,\ roots,\ commitThreshold,\ bondedWithdrawals \rangle$

$CommitTransfers(c)\ \triangleq$
$\land\ SumSeq([x \in \text{DOMAIN}\ pendingTransfers[c] \mapsto pendingTransfers[c][x].amount]) > commitThreshold$
$\land\ Print(\langle \text{“commit”} \rangle,\ \text{TRUE})$
$\land\ pendingTransfers' = [pendingTransfers\ \text{EXCEPT}\ ![c] = \langle\rangle]$
$\land\ roots' = [roots\ \text{EXCEPT}\ ![c] = roots[c] \circ \langle Hash(pendingTransfers[c]) \rangle]$
$\land\ roots' = [roots\ \text{EXCEPT}\ ![l1Chain] = roots[l1Chain] \circ \langle Hash(pendingTransfers[c]) \rangle]$
$\land\ \text{UNCHANGED}\ \langle l1Chain,\ l2Chain,\ commitThreshold,\ bondedWithdrawals \rangle$

$BondWithdrawal(c)\ \triangleq$
$\land\ Len(pendingTransfers[c]) > 0$
$\land\ \exists\, x \in \text{DOMAIN}\ pendingTransfers[c] :$
$\land\ pendingTransfers[c][x].id \notin bondedWithdrawals[c]$
$\land\ Print(\langle \text{“bondWithdrawal”} \rangle,\ \text{TRUE})$
$\land\ bondedWithdrawals' = [bondedWithdrawals\ \text{EXCEPT}\ ![c] = bondedWithdrawals[c] \cup \{pendingTransfe$
$\land\ \text{UNCHANGED}\ \langle l1Chain,\ l2Chain,\ commitThreshold,\ pendingTransfers,\ roots \rangle$

$Next\ \triangleq$
$\land\ \exists\, i \in 1\,..\,2 :$
$\land\ \exists\, c \in l2Chain :$
$\land\ \lor\ SendTransfer(c)$
$\lor\ CommitTransfers(c)$
$\lor\ BondWithdrawal(c)$

$Spec\ \triangleq\ Init \land \Box[Next]_{\langle l1Chain,\ l2Chain,\ pendingTransfers,\ commitThreshold,\ roots,\ bondedWithdrawals \rangle}$

$pendingTransfers[chainid] = \langle 1,\ 2 \rangle$