```
- MODULE HopProtocol -
EXTENDS Integers, Naturals, TLC, Sequences, FiniteSets
CONSTANT
  B,
               block
   CT
               commit threshold
VARIABLES
             chain
Pick(S) \stackrel{\triangle}{=} CHOOSE \ s \in S : TRUE
RECURSIVE SetReduce(_, _, _)
SetReduce(Op(\_, \_), S, value) \stackrel{\triangle}{=}
  If S = \{\} then value
   ELSE LET s \stackrel{\triangle}{=} Pick(S)
            IN IF Op(s, value) = Op(value, s)
             THEN SetReduce(Op, S \setminus \{s\}, Op(s, value))
             ELSE Assert(FALSE, "Err")
SumSeq(S) \triangleq
  Let seq \triangleq S
          Sum[i \in 1 ... Len(seq)] \stackrel{\triangle}{=} IF i = 1 THEN seq[i] 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 \stackrel{\triangle}{=} CHOOSE x \in S: TRUE
            IN \langle x \rangle \circ SeqFromSet(S \setminus \{x\})
Hash(v) \stackrel{\Delta}{=} CHOOSE \ n \in \{Int\} : TRUE
ASSUME (Hash(\langle 1 \rangle) = Hash(\langle 1 \rangle))
ASSUME (Hash((\{1, 2, 3\})) = Hash((\{2, 1, 3, 1\})))
TypeOK \triangleq
   \land\, B \subseteq 1 \mathinner{.\,.} 5
   \land \exists c \in 1 \dots 2 : C[c].h \in 1 \dots 5
Init \stackrel{\triangle}{=}
   \land \ C = [c \in 1 \dots 2 \mapsto [
       id \mapsto c,
                                         {\rm chain}\ id
       h \mapsto 1,
                                         block height
       pt \mapsto \langle \rangle,
                                         pending transfers
       pts \mapsto 0,
                                         pending transfers sum
       tr \mapsto 0
                                         transfer root
```

```
]]
Add(c) \triangleq
   Let s \triangleq \langle (RandomElement(1..2)) \rangle \circ C[c].ptin \quad s
Send(c) \triangleq
    \wedge Print(\langle C \rangle, TRUE)
    id \mapsto C[c].id,
           h \mapsto C[c].h,
           pt \mapsto Add(c),
           pts \mapsto SumSeq(Add(c)),
           tr \mapsto C[c].tr
      ]]
 Commit(c) \triangleq
    \land \ C[c].pts > CT
    \wedge C' = [C \text{ EXCEPT } ! [c] = [
          id \mapsto C[c].id,
           h \mapsto C[c].h,
           pt \mapsto \langle \rangle,
           pts \mapsto 0,
           tr \mapsto C[c].pts
      ]]
\textit{Next} \; \stackrel{\Delta}{=} \; \land \, \exists \; bn \, \in \, B :
                   \lor \land \exists c \in \text{domain } C:
                           \land \lor Send(C[c].id)
                               \vee Commit(C[c].id)
                   \vee \, \wedge \, C' = [c \in \text{domain } C \mapsto [
                               id \mapsto C[c].id,
                               h \mapsto bn,
                               \begin{array}{ccc} pt & \mapsto C[c].pt, \\ pts & \mapsto C[c].pts, \end{array}
                               tr \mapsto C[c].tr
                          ]]
Spec \stackrel{\Delta}{=} Init \wedge \Box [Next]_{\langle C \rangle}
```