
MODULE *Blockchain*

This module defines network initial conditions to be used by the *p2p* algorithm.

EXTENDS *Integers, Sequences, TLC, Utils*

Create a network with given number of peers, the blocks and connections to be established.

$CreateNetwork(numPeers, blockCounts, connections) \triangleq$

$$\begin{aligned}
 & [peer \in 1 \dots numPeers \mapsto \\
 & \quad LET \ numBlocks \triangleq \ blockCounts[peer] \\
 & \quad \quad \text{Construct } peer_set \text{ as a sequence of other peers, seeder nodes have no connections.} \\
 & \quad \quad peerSet \triangleq IF \ connections[peer] = TRUE THEN \\
 & \quad \quad \quad Remove(\\
 & \quad \quad \quad \quad \text{Add all peers to the list.} \\
 & \quad \quad \quad \quad [i \in 1 \dots numPeers \mapsto [\\
 & \quad \quad \quad \quad \quad address \mapsto \text{"peer"} \circ ToString(i), \\
 & \quad \quad \quad \quad \quad tip \mapsto blockCounts[i], \\
 & \quad \quad \quad \quad \quad established \mapsto FALSE \\
 & \quad \quad \quad \quad]], \\
 & \quad \quad \quad \quad \text{Remove the current peer from the list ...} \\
 & \quad \quad \quad \quad [\\
 & \quad \quad \quad \quad \quad address \mapsto \text{"peer"} \circ ToString(peer), \\
 & \quad \quad \quad \quad \quad tip \mapsto blockCounts[peer], \\
 & \quad \quad \quad \quad \quad established \mapsto FALSE \\
 & \quad \quad \quad \quad] \\
 & \quad \quad \quad ELSE \langle \rangle \\
 & \quad IN \ [\\
 & \quad \quad peer \mapsto \text{"peer"} \circ ToString(peer), \\
 & \quad \quad blocks \mapsto ToSet([height \in 1 \dots numBlocks \mapsto [\\
 & \quad \quad \quad height \mapsto height, \\
 & \quad \quad \quad hash \mapsto \text{"blockhash"} \circ ToString(height), \\
 & \quad \quad \quad block \mapsto \text{"serialized block data"} \circ ToString(height) \\
 & \quad \quad]]), \\
 & \quad \quad peer_set \mapsto peerSet \\
 & \quad] \\
 &]
 \end{aligned}$$

2 peers network. 1 seeder with 1 block and no outbound connections and 1 peer with no blocks and an outbound connection to the seeder.

$Blockchain1 \triangleq CreateNetwork(2, \langle 1, 0 \rangle, \langle FALSE, TRUE \rangle)$

2 peers network. 1 seeder with 10 blocks and no outbound connections and 1 peer with no blocks and an outbound connection to the seeder.

$Blockchain2 \triangleq CreateNetwork(2, \langle 10, 0 \rangle, \langle FALSE, TRUE \rangle)$

3 peers network. 1 seeder with 1 block and no outbound connections and 2 peers with no blocks and an outbound connection to the seeder.

$Blockchain3 \triangleq CreateNetwork(3, \langle 1, 0, 0 \rangle, \langle FALSE, TRUE, TRUE \rangle)$

1 peer with or without connections or blocks is an assert

$Blockchain4 \triangleq CreateNetwork(1, \langle 0 \rangle, \langle TRUE \rangle)$

2 or more peers without connections is a deadlock

$Blockchain5 \triangleq CreateNetwork(2, \langle 0, 0 \rangle, \langle FALSE, FALSE \rangle)$

2 peers network. 2 connected to each other with the same amount of blocks.

$Blockchain6 \triangleq CreateNetwork(2, \langle 100, 100 \rangle, \langle TRUE, TRUE \rangle)$

2 peers network. 2 connected to each other with different amount of blocks.

$Blockchain7 \triangleq CreateNetwork(2, \langle 2, 1 \rangle, \langle TRUE, TRUE \rangle)$