ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2016 Prof. R. Wattenhofer

Distributed Systems Part II

Exercise Sheet 6

Quiz

1 Is delayed Bitcoin strongly consistent?

In the lecture we have seen that Bitcoin only has eventual consistency guarantees. The state of nodes may temporarily diverge as they accept different transactions and consistency will be re-estalished eventually by blocks confirming transactions. If, however, we consider a delayed state, i.e., the state as it was a given number Δ of blocks ago, then we can say that all nodes are consistent with high probability.

a) Can we say that the Δ-delayed state is strongly consistent for sufficiently large Δ?

b) Reward transactions make use of the increased consistency by allowing reward outputs to be spent after *maturing* for 120 blocks. What are the advantages of this maturation period?

2 Doublespending

Figure 1 represents the topology of a small Bitcoin network. Further assume that the two transactions T and T' of a doublespend are released simultaneously at the two nodes in the network and that forwarding is synchronous, i.e., after t rounds a transaction was forwarded t hops.

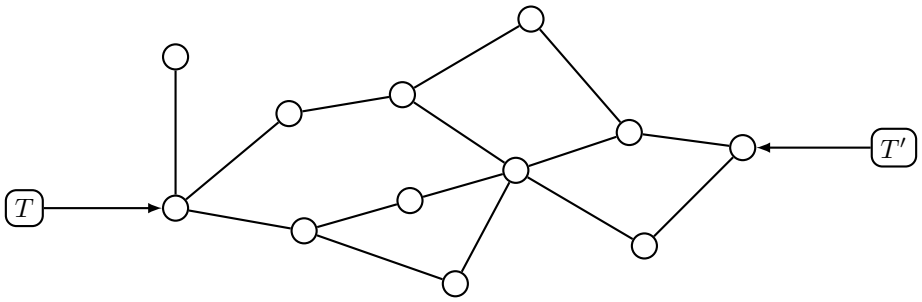

Figure 1: Random Bitcoin network

a) Once the transactions have fully propagated, which nodes know about which transactions?

b) Assuming that all nodes have the same computational power, i.e., same chances of finding a block, what is the probability that T will be confirmed?

c) Assuming the rightmost node, which sees T' first, has 20% of the computational power and all nodes have equal parts of the remaining 80%, what is the probability that T' will be confirmed?

3 Partially spending outputs

As seen in the lecture, inputs claim the entire value associated with an output, even if the intended transfer is for a much smaller value than what the input references. If the input claims a larger value than needed for the transfer the user simply adds a *change output*, which returns the excess bitcoins to an address owned by the sender.

Why do inputs always spend the entire output value and not just the part that is needed for the transfer?

4 Replacement using sequence numbers

The original Bitcoin protocol included sequence numbers which in combination with timelocks allowed replacing transactions with newer versions. Each input is assigned a 4 byte sequence number which describes the precedence in which transactions should be considered. If a node receives two transactions with inputs referencing the same coins, then the node should prefer the transaction with the higher sequence number. However, this replacement has been disabled because it did not work reliably and enabled DoS attacks against the network.

a) Why did the replacement not work reliably and how could it be misused to attack the network?

b) Given two transactions, is it always clear which transaction should be prioritized?