Project Overview

Shoma Mori

October 22, 2019

1 Nodes

There are two types of nodes in the system, namely, server and client. Servers are basically validators, who verify transactions. Clients are nodes who issue transactions. Servers behave independently and so do clients, which means the system is asynchronous.

2 Transactions

The following is the flow that describes how a transaction from client c_1 to c_2 is approved by the system without a consensus.

1. c_1 creates a transaction that indicates a payment from c_1 to c_2 using UTXO linked to c_1.
2. c_1 sends the transaction to all servers.
3. The servers verify the transaction by checking whether there are no conflicts with the history of transactions in their local storage.
4. If not, the servers send c_2 the transaction with their own signatures which were generated from their private keys and add the transaction to the local storage.
5. When c_2 receives the transaction, it verifies the signatures using the corresponding public keys.
6. When c_2 receives valid signatures from more than two-thirds of all nodes, it regards the transaction to be approved.

3 Double-spending

Consider the situation that one client sends transactions that imply double-spending before none of them are not approved (Fig. 1). If a server receives more than one transaction of them, it can obviously detect double-spending. If a server receives only one of them (meaning that the client tried so), the server may sign that transaction. However, since the clients cannot send more than one transaction to 2/3 of all nodes without overlapping, more than one transaction cannot be approved. In that case, only one of them would be approved in the best case, and no transactions are approved in the worst case.

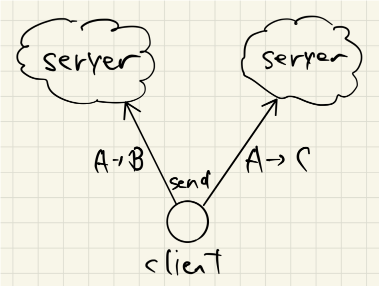

Figure 1: Double-spending

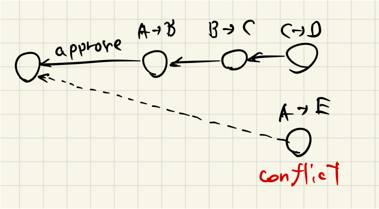

Figure 2: Double-spending

Next, consider that one client sends conflicting transactions after the conflicted transaction is approved (Fig. 2). In this case, servers can detect double-spending easily by checking their local storage and finding the conflicting transactions.