Zero Reserve – A Distributed Exchange Platform

(v. 0.1)

Draft – do not distribute

Rüdiger Koch

Abstract: A P2P market place for Bitcoin would permit trading and price discovery even in the absence of cooperation from the legacy payment infrastructure. Orders get signed and published through the network and matched by each node. A buyer sends a Ripple-like payment to the seller who sends Bitcoins as part of one single transaction. The system requires trust in friends, but no further.

Introduction

Distributed P2P cash systems like Bitcoin have been a wild success. Yet a weakness remains: they are entirely dependent for trading and thus, for price discovery on centralized exchanges and marketplaces. These platforms are dependent on the cooperation of the competition, however, namely the banks. Without the ability to wire fiat money into and out of the exchange, the current markets cannot exist. In addition, the existing market places are more and more encumbered with breaches of privacy by local and foreign sovereigns.

What is needed is a system that does not require the legacy banking system to move money in and out of the system, and a distributed, anonymous order book that allows for transparent transactions. A system that fulfills the first requirement is known – the original Ripple idea permits everyone who is deemed trustworthy by his friends to assume the role of money creation – a role which is traditionally reserved for the banks. A second requirement is a distributed order book and distributed order matching. Lastly, such a system needs to implement transactions, where an unknown peer is payed and Bitcoins are transferred reliably as part of the transaction. In this paper, such a system is proposed.

Ripple

Ripple is a payment system first introduced by Ryan Fugger which takes the idea of Local Exchange Trading Systems (LETS) a step further to permit arbitrary scalability. It is based on credit given to friends. A Ripple friend is someone who is trusted not to default on the credit granted.

If a friend grants you a credit, you are able to pay him up to that amount. A remote payment to people who are not friends and who might be completely unknown is based on finding a chain of friends-of-friends-of-friends..... until the chain reaches the payee. It is based on the finding of social network theory that the length of such chains is usually quite limited – the number of hops between any two randomly chosen people (anecdotally) rarely exceeds 6.

Bitcoin

Bitcoin is a P2P distributed electronic payment system which relies on an append-only database of transactions of which each full node holds a copy. A transaction is the transfer of “coins” between one address to another in this database (block chain). It is signed by one or more private keys, belonging to one or more owners.

The Order Book

Matching of new bids and asks is first tried locally. If the order matches partially the part that matches is executed and the adjusted order is published. If it matches fully, it is executed and nothing is published. Orders are published to all friends, who republish them to all of their friends. If a node already has an order it receives, it does not re-publish it.

Orders that are partly matched are published again, with the adjusted amount. Orders that are fully matched are published with a notification for nodes to remove this order from their book.

As a protection against order spamming (an implicit DOS attack) publishing a new order costs a very small amount in Bitcoins, to be payable to an organization of the publisher's choice. To that end, a Bitcoin micropayment channel is opened to that organization.

In addition to the order information, published orders contain proof of pay, a hash that permits a counterparty to contact the publisher, a public key so the counterparty can encrypt communications so it can't be read by intermediate nodes and a signature to make it tamper proof.

Credit

When connecting to a friend, a user can choose to grant credit. The friend can then commit to transactions with that user up to his credit line, in the currency in which the credit was granted.

Available funds are credit minus debt minus reserved funds for ongoing transactions.

Transactions

Three types of transactions are common:

1. **Simple payments to friends.**
2. **Bitcoin purchase transactions.**
3. **Triangular Debt cancellation transactions.** This is a maintenance transaction to optimize the performace of Zero Reserve.

Simple Payments to friends

The trust relationship between payer and the payee and the fact that they are directly connected with no hops permits for a simple 2-phase commit protocol (2PC). The 2PC is not strictly atomic which can lead to inconsistent records at the payer and payee side in rare cases. This is acceptable here because such payements will always explicitly be agreed upon by the payer and payee.

In case such a payment leads to a commit on only one side, the inconsistency is discovered latest when the 2 nodes compare their mutual debt record at which point both nodes will notify their users who will then need to manually fix the situation, usually by a cancellation of the committed payment on one side and a second attempt to pay.

This payment is mainly used for settlement. For example if one friend exhausted his credit by

buying a lot of Bitcoin he will do a cash payment or bank transfer to settle that. A simple payment is then done to reflect that in the Zero Reserve book.

Bitcoin purchase payment

Bitcoin purchases uses a modified 2PC which takes into account that the coordinator of a payment (always the fiat payer / Bitcoin buyer) is only connected directly to the next hop.

In the first phase, the coordinator sends a QUERY to the payee (the Bitcoin seller) by selecting a route via a friend who is closer to the payee. This QUERY is accompanied by the bitcoin amount, the price and the destination Bitcoin address. Hops either pass on the QUERY or returns a NO vote. In case of a NO, the transaction is cancelled. Unlike the standard 2PC, subsequent hops will not be notified of the vote because they did not yet get any information about the ongoing TX. Any hop, including the payee, can lower the amount of a trade if credit is insufficient. Insufficient credit on a hop is not a reason to cancel the TX.

The payee then returns the trade with a YES vote, potentially an updated amount and a Bitcoin TX ID. Each hop enters a binding agreement that if the TX with that ID reaches 6 confirmations on the Bitcoin network, a payment is done, once they get the final COMMIT from the coordinator/payer, as they pass on the YES vote. Funds are reserved for this TX from the available credit until the TX either times out, is cancelled by the coordinator or the contract is fulfilled.

When the coordinator receives the contract and agrees to it, he sends the COMMIT. When the payee receives the COMMIT, he sends out the Bitcoin transaction to the network. The COMMIT only means a commitment to the contract, not a commit of the payment.

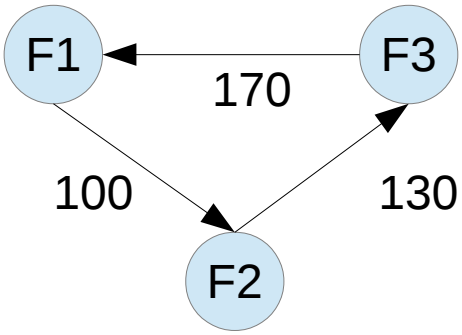
Every party in the contract then polls the Bitcoin network for this Bitcoin TX until the 6 confirmations are reached. Then they commit the payment to their books without any further communication between the.

Should the final COMMIT for some reason never reach the Bitcoin seller, his fiat TX will time out and the Bitcoin TX is not sent to the Bitcoin network and thus not incorporated into the blockchain. In that case, all hops will time out this contract after 24 hours. The same happens if the TX is sent but does not satisfy the contract, for example if the Bitcoin amount is too low or the destination address does not match the one sent by the Bitcoin buyer in the QUERY phase.

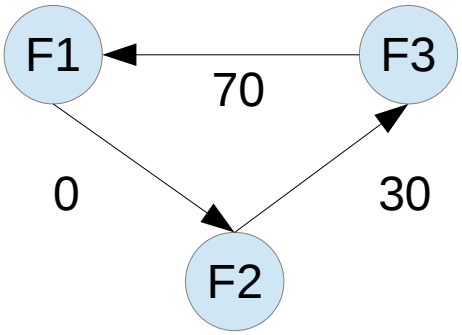
Neither the buyer nor the seller are known even by the next hop, providing a very high level of anonymity. This anonymity offers a gap for a DOS attack: A node can initiate transactions, but drop them before the commit happens. It should be noted that this can be done only until the reserved funds of the next nodes reach their limit. When too many TX are ending up timed out, nodes should notify their user that a friend is possibly an abuser.

Triangular Debt cancellation

Frequently, situations will arise where 3 mutual friends owe each other in such a way that debt can be cancelled out to reduce each one's debt:



With 3 simple transactions combined in one, where 100 is transferred from F1 → F3 → F2, the debt is now:



Transaction Fees

Intermediate hops can impose a transaction fee. This is an incentive to participate in transaction processing. The negotiation of fees need to preceed the actual transaction so that the amount + accumulated fees can be queried from each hop.

As a payment may select any of multiple routes, competition for routing will keep fees reasonable.

The fees are retained if the transaction is aborted by the payer after the all the nodes voted “yes” or if it times out after that. This is important to prevent attacks on the network with the aim to create many escrow cases.

Routing

To send payments via intermediates, it is necessary to find routes from the payee to the payer to the payee. Zero Reserve uses a variation of turtle routing where friends are eliminated from a route if they do not fulfil the requirement to route a payment.

It is a common case that a payment is too large for any single route. Additional routes must then be found to fulfil the payment requirement.

Implementation

Zero Reserve is designed as a Retroshare Plugin. It uses the underlying Friend-to-Friend model,

communication infrastructure, security model and to an extent, turtle routing of Retroshare.

References

Retroshare Website: http://retroshare.sourceforge.net/

Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System

Ryan Fugger: A Proposal for a Secure, Private, Decentralized Digital Currency Protocol

Popescu, Crispo, Tanenbaum: Safe and Private Data Sharing with Turtle:
Friends Team-Up and Beat the System