

# Zero Reserve – A Distributed Exchange Platform

anu  
anu@zerostate.net

**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, price discovery on centralized exchanges and market places. 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 fulfils 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 arbitary 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 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 ability to sign a transaction with multiple keys is an integral part of transactions in Zero Reserve.

## The Order Book

New bids and asks are first tried to match locally. If the order matches partly 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 which could take the character of a DOS, 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.

## Transactions

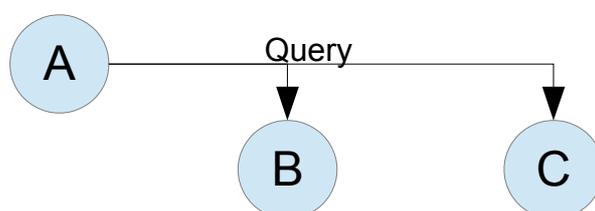
Three types of transactions are common:

1. **Simple payments.** Use case for this is when the other side of the transaction is outside of Zero Reserve, such as paying back the debt by using a bank transfer.
2. **Bitcoin purchase transactions.**
3. **Triangular Debt cancellation transactions.** This is a maintenance transaction to optimize the performance of Zero Reserve.

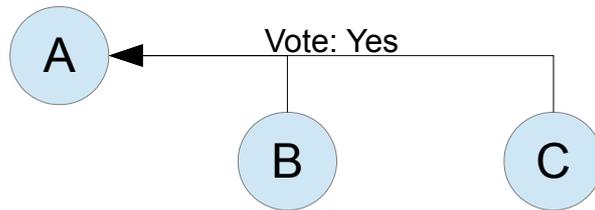
The following explanations assume a 2-phase commit protocol although the actual implementation may deviate from this.

### *Simple Payments*

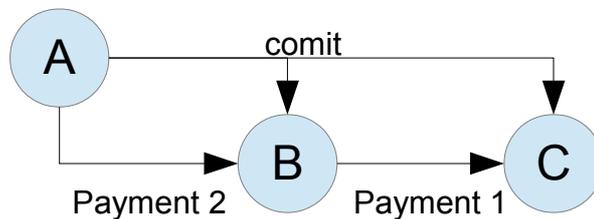
To simplify, consider a payment of a stranger over one hop i.e. a friend-of-friend. Of course more hops are possible. The payer is the coordinator of the transaction. In the first stage, the payer (A) queries the hops (B) and the payee (C) if they are willing to partake in the transaction of the given amount.



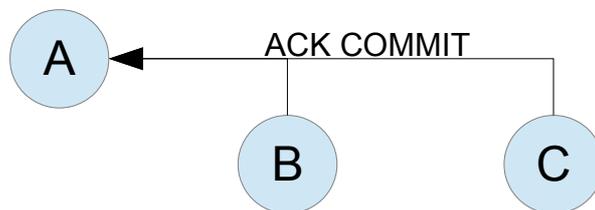
If all parties agree, they vote “yes”. If one party does not agree, it votes “No” and the entire transaction is aborted. If the reason is insufficient funds, the node may return a proposal for a devised transaction with the “No” vote. This way, the complete payment may be composed from payments along different routes.



The next stage is the commit where the chain of payments is done. The chain of payments is backwards – the payee is paid first (payment 1), the payer pays last (payment 2). This way, the chain of trust is maintained – it is only possible to break the trust of a friend without gain.



Each node that is finished sends an acknowledgement back to the payer.



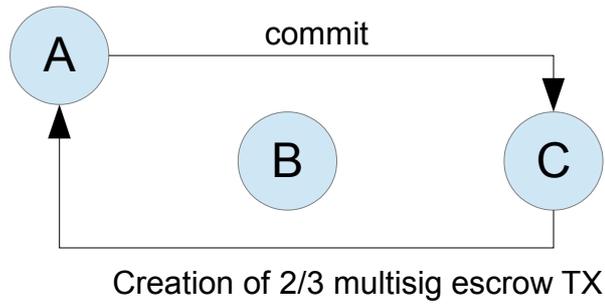
Two kinds of payments are possible:

1. A guaranteed mode, where a single route must be found to transfer all in a single transaction. Should no such route be found, the payment fails and nothing is done.
2. In the best effort mode, a payment can be split along multiple routes, but those multiple transactions are then all independent: The payment is not “all or nothing” - if all routes that can be found offer less capacity than the planned payment requires, the payment will be done up to that capacity.

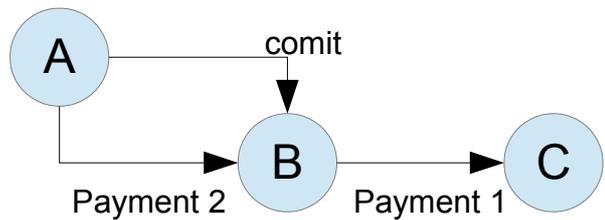
A guaranteed mode where a transaction is split but all transactions are atomic is possible, but needs to be researched. For example such transactions may fail very often, because the number of involved nodes is very high.

### ***Bitcoin purchase payment***

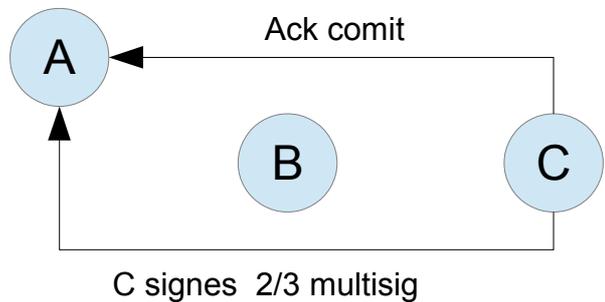
A Bitcoin purchase payment adds a 2/3 escrow transaction to the procedure of a simple payment. The commit signal goes to the payee (C) first who creates the transaction:



Then the payer issues the remaining commits.



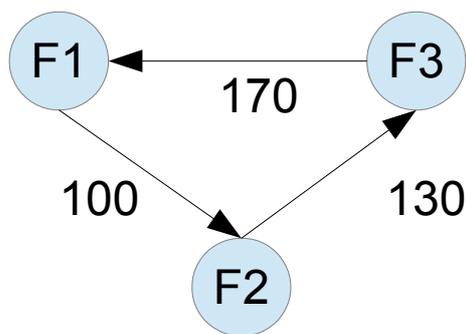
At the acknowledgement phase, the signature is given by the payee (C) so A can now spend the coins.



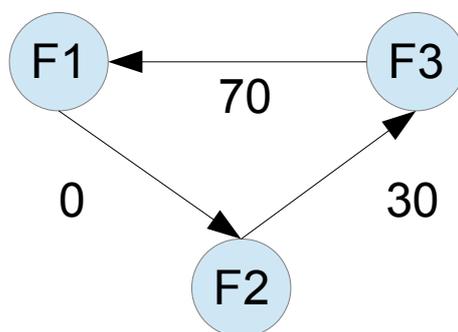
This would leave the network open to an attack where a buyer initiates large numbers of transaction which he abandons after the escrow transaction is created, leaving them all to the escrow to resolve. As the transaction fee is not redeemable for transactions that are abandoned after the vote, this attack would be costly for the attacker.

### ***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  $F1 \rightarrow F3 \rightarrow F2$ , 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 precede 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 extend, 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