Introduction

How turbo tx work

How traditional 0-conf tx work

Traditional 0-conf tx work by allowing a user to accept a tx that has not been added to a block yet. These are not guaranteed tx, but are relative low risk. Based on this system its said that 0-conf tx are safe to a point for small amounts (like buying small products from a store) where the wait time needs to be much lower than the block time. this is based on a few things:

#1 All tx should be moved into the next block (the faster the tx is added the better). This is due to the longer a tx is in the mem pool (the waiting area for tx before they go into a block) the more chance their is for that tx to get reversed by the sender.

#2 The only way to replace a tx is to, send a tx to the store, buy the products, leave the store and send another tx to yourself, then have a modified daemon that will accept your second tx and control enough network power (does not have to be 51% can be 25% for a 1/4 chance etc) to have your daemon mine the block and insert the second tx.

Since most proof of work coins have only a few centralized pools, the pool operators can have a high chance to reverse transactions since all workers are sending hashes to their daemon. Any one accepting 0-conf tx on this kind of network have these 2 aspects to trust that wont happen.

This is more a less a convenience thing. a store can accept 0-conf on the off basis this will happen, but for small style line waiting purchases. Obviously if buying a car or something it would be best to wait for at least one confirmation, as only a 51% attack could reverse it at that point.

With xcash's unique combination of unspent based blockchain and DBFT consensus mining, their are ways to improve the current 0-conf tx disadvantages.

Do 0-conf tx work on all blockchain

For the most part no, their is only a select combination of blockchains they can work on. Most account based blockchains (where they use a nonce to index txs and only store your current balance) wont work. This is due to one could just send the tx to the store, and then quickly send another tx with a higher fee to themselves and cancel out the store tx once in their car. Only blockchains that are unspent based (where it keeps a record of all the in and out amounts that make your balance) can work. This is due to because daemons on the network will reject double spent unspents, or unspents that are already in the mem pool (waiting to be added to a block). However some blockchains like Bitcoin have a replace by fee (even though bitcoin is a unspent style blockchain), so it would not work on this type of unspent blockchain.

How are xcash turbo tx different than traditional 0-conf tx

In xcash things work differently due to xcash DPOPS. Only the top 50 delegates can mine blocks on the network. One can have their own delegate in the top 50 and attempt a replace but they would need to be picked so they only max can have a 2% chance. This right here is more decentralized than most proof of work coins. But their are still ways to make this better, and xcash has figured out how to make guaranteed 0-conf tx.

Here is how this works xcash DPOPS works off of DBFT (delegated byzantine fault tolerance). DBFT requires everyone to verify a block **BEFORE** adding it to the network. Because of this if a delegate modified its xcashd to accept that second tx (the fake or replacement tx), it wont show in the mempool for the other 49 delegates.

Also a valid tx wont be in all 50 delegates, but should be in the majority more than 27. Because of this if a DBFT check of each individual transaction that the block producer selected to mine is run, their will never be a reversed tx being added to a block based on these few rules above, as the other 49 will always stop the malfunctioning tx and or delegate. As long as 27+ DPOPS delegates are true, this holds. If not their are way worse things to worry about so its a non point to look at what if not. (For example delegates could just mine infinite blocks every few seconds etc etc)

if a user uses a xcashd that has the 0-conf tx in the mempool (or can find a delegate who has it) and verifies the tx pub key and outputs, then 0-conf tx are confirmed valid before they are added to the block.

This would 100% eliminate the concerns for 0-conf. This is because if 27+ delegates mempool have a tx in it, it will be mined at some point due to the DBFT and unspent features. Also if a delegate has invalid tx it wont be mined due to the dbft check.