Inside Ethereum Transactions

Author	josh
Date	2019-05-16 17:00:19

Overview

Much like Bitcoin, Ethereum is a wildly popular cryptocurrency that allows its users to exchange value on a global, decentralized, censorship resistant platform. However, Ethereum goes beyond the capabilities of Bitcoin by adding a far more versatile scripting platform for the creation of "smart contracts" or "decentralized applications".

In order to facilitate this, the Ethereum blockchain uses a different model for tracking ownership of funds than the Bitcoin blockchain. Bitcoin, Bitcoin Cash, Litecoin, and other popular currencies are UTXO based blockchains. Ethereum, however, uses an account based model. Let's take a closer look at Ethereum's model for dealing with address balances and transactions.

The Ethereum Model

UTXOs vs. Accounts

On the Bitcoin blockchain, exchanges between addresses are facilitated by consuming and creating "Unspent Transaction Outputs", or UTXOs. When someone creates a transaction to send value to another address, their wallet looks for enough UTXOs to use an inputs for the transaction. Each UTXO used must be consumed in full, much like a dollar bill or cent coin in fiat money. The transaction is then constructed with outputs (new UTXOs) owned by the recipient. If there's extra money that had to be consumed from input UTXOs, the user's wallet will send change back to itself in the form of an output.

It's just like going to the store and buying a $15 item with a $20 bill - you give the cashier the $20 and they give you $5 back in change. This UTXO model behaves remarkably similar to physical cash.

Ethereum, however, uses an account-based model that behaves much more like a traditional bank account. Each address is considered an "account", and each transaction marks a change in state on that account. When an Ethereum transaction is created, the input account has a negative change in value (a withdrawal). The output account receives a positive change in value (a deposit). There's no bills exchanged here, there's simply an electronically recorded change in the state of the account's balance.

Types of Ethereum Accounts and Transactions

The reason Ethereum uses an account model becomes more clear considering its more advanced scripting capabilities than Bitcoin. There are two different types of accounts in Ethereum: **Externally Owned Accounts** and **Contract Accounts.**

Contract accounts are created when a new smart contract is deployed on the network - these accounts are only controlled by the code of that contract. EOA accounts are the accounts we are focused on for understanding value transfers between individuals - when you give someone your Ethereum address or send to someone else's, you're dealing with Externally owned accounts.

This leads us to the few types of transactions in Ethereum. There's transactions that **deploy smart contracts** by providing the contracts code to the network. There's transactions that s**end messages to contracts in order to trigger execution**. And finally, there are **simple value-transfer transactions conducted between externally owned accounts**. Again we are focused here on this type of transaction.

Inside an Ethereum Transaction

Now, let's look a bit deeper at the transaction internals in order to fully understand how an Ethereum transfer works. Here's an example of a simple value transfer between to addresses. I've simplified and reformatted this data grabbed from a block explorer API to make it easy to understand:

```
{ "transaction_id":
"7959c9b9c9f0e949e6fa175d8b15d3bb464d31659f8c607d43b59c96e9bffe47",
  "total": 0.17703882,
  "fees": 0.00042000,
  "gas_limit": 21000,
  "gas_price": 0.00000002,
  "from": "fd7079c59b403759264d477c7b71105be0319de8"
  "to" : "e7e485512e0c2b7b21f7bad2d43fb83bce2886e4"
  "nonce": 0
}
```

Let's discuss each piece individually:

Transaction ID - A hash of the transaction data that uniquely identifies the tx on the blockchain
Total - The amount of Ether transacted in total
Fees - Ether paid to the miner to incentivize the inclusion of the transaction in a block
Gas limit - Gas is a unit that describes the cost of each computational step when executing a transaction/contract. Each computation requires a certain amount of gas to execute, and this is the *maximum* amount of gas the transaction creator is willing to pay.
Gas price - The price per computation (like dollars per gallon of real-life gasoline)
From - The account sending the Ether, whose account balance will be reduced
To - The account receiving the Ether, whose account balance will be increased
Nonce - A value that increases for every transaction the sender completes from their account. It's used to help track state over time and prevent double spends from an account.

Ethereum Transactions - Like Your Bank Account!

All in all, this model seems a bit simpler to understand than the BTC/BCH/LTC UTXO model, but it also facilitates more complex blockchain use since accounts can be used for everything from basic transfers to contract deployment and execution.

When Ethereum transactions are executed on the blockchain, what you're really seeing is a change in the *state* of the address (account). The sending account has its balance reduced, and the receiving account has its balance increased, just like a bank transfer.

Other transaction components like fees and gas are also fairly intuitive - these are a premium paid by the sender to allow their transaction to be included in a block, and a premium paid for the privilege of using the miner's computational power. Unlike a traditional bank, this system allows account transfers in a manner that doesn't require any trust!