The Architecture Of A Cryptocurrency

Though we have thousands cryptocurrencies buzzing around at the moment, all of them share the same implementation design. Basically every cryptocurrency implementation has following parts:

* Network Protocol
* Consensus Protocol
* Transaction Protocol
* Internal State

Network Protocol

P2P network is the core of a cryptocurrency, and the core is following some network protocol. It is pretty simple usually. For example, p2p network of a cryptocurency could be implemented with asynchronous exchange of the following messages:

* M1: Serialized unconfirmed transaction
* M2: Serialized block
* M3: Blockchain quality score request(e.g. height for the Bitcoin, cumulative difficulty for the NXT).
* M4: Blockchain quality score response
* M5: Get known peers request
* M6: Get known peers response
* M7: Request a block for a certain height(or with a certain parent)

The minimalistic set is pretty similar to Scorex message types. Bitcoin has more complicated protocol for sure(https://en.bitcoin.it/wiki/Protocol_documentation#Message_types) as well as NXT.

Let's consider an example of a blockchain downloading with messages types given above if peer A knows only peer B and has genesis block only. So possible interaction of A and outer world where A -> (B, M2) means (peer A sends an instance of message kind M2 to peer B):

1. A asks B for peers: A -> (B, M5)
2. A asks B for a B's blockchain height: A -> (B, M3)
3. B replies for the first message with known peers C & D: B -> (A, M6)
4. B replies for the second message with it's height H(B): B -> (A, M4)
5. A asks C for its blockchain height: A -> (C, M3)
6. A asks D for its blockchain height: A -> (D, M3)
7. C replies with its blockchain height H(C): C -> (A, M4)
8. D replies with its blockchain height H(D): D -> (A, M4)

So A have 3 known heights H(B), H(C), H(D) and a bigger height means better chain(for a Proof-of-Work cryptocurrency), so A chooses peer with a best chain, e.g. D. Then it's going to download a chain from D with a following cycle:

9. A asks D for a block with height (2,3,...,H(D)): A -> (D, M7)
10. D replies with a block requested serialized into a binary form: D -> (A, M2)

Any real trace is much more complicated than this example :)

Consensus Protocol

By using network protocol a node can download one chain or another or even multiple chains(as different peers can have different chains). Then the question is what is the canonical chain shared by majority, the right kind of history an user can rely on. Consensus protocol aims to solve the problem.

A node follow rules to determine canonical chain like these:

1. Every block in a chain must be valid as well as its signature

2. Every block must have a valid reference to its parent with first block having reference to a genesis block which is constant for an each node

3. Every block must be generated by a party having a right to produce it

4. From multiple chains, one having maximum score is the canonical. If few chains have the same score first seen or a random one could be chosen

Some notes about the rules:

* Block validity fact could be changed with a new software release. Most known example of that is Bitcoin network splitting caused by 0.8 release

* In proof-of-work cryptocurrencies a party has a right to generate block if hash of its contents conforms to some condition(its cheap to calculate a hash and compare it with some value while it's hard to iterate over nonce bytes to find such a hash). In proof-of-stake cryptocurrencies a party has verifiable right to generate a block if some pseudo-random value `hit` is less than stake-dependent `target`. It's harder to design safe Proof-of-Stake cryptocurrency, for example, majority of such currencies are vulnerable to grinding attack(Nxt is not).

* In Bitcoin "longest chain rule" being applied while in Nxt there's another blockchain score function called "cumulative difficulty"

* More details on the Nxt algo: [http://chepurnoy.org/blog/2014/10/inside-a-proof-of-stake-cryptocurrency-part-2/](http://chepurnoy.org/blog/2014/10/insid

Transaction Protocol

A block contains transactions. A transaction is the global state modifier. In simplest case it modifies balance sheet. As there's no central party, each node need to have the same state as others(for the same block height). So each node executes all the transactions coming within blocks, and validity rules must be same for each network participant.

A transaction could be implemented in different ways depends on goals:

* In Bitcoin, transaction has multiple inputs and outputs with a script attached to every pin.

* Ethereum has two transaction types, contract creation and message call. See the Yellow Paper for details:
http://gavwood.com/Paper.pdf.

* In Nxt, transaction has some money amount and attachments. There are some rules on attachments processing,
e.g. some kind of attachment means plain or encrypted message while other means order on
decentralized exchange. Such an implementation provides maximum performance for features but has own
disadvantages(complicated code, hardforks on each software update).

Internal State

In Bitcoin node a can store no any state at all or use different formats for it, e.g. UTXO list or some indexes in addition.
In contrast, Ethereum has state hash stored into each block, so each node must comply with state representation
interface given in the Yellow Paper to check block validity. There's also interesting proposal from Bill White(coming
along with Coq theory) about using special kind of Merkle tree and storing its root into block for scalability's sake:
[http://qeditas.org/gitweb/?p=ledgertheory.git;a=blob_plain;f=lightcrypto.pdf;hb=HEAD]
(http://qeditas.org/gitweb/?p=ledgertheory.git;a=blob_plain;f=lightcrypto.pdf;hb=HEAD).

The Protocols Mess Problem

The real problem of cryptocurrency development process is the mess of all the parts described above.
The most known example, even simple change in software validation(so in Transaction Protocol) could causes
hardforking.

Acknowledgments

The idea of Network/Transaction/Consensus protocols was initially explicitly stated by L.M Goodman in the
"Tezos: A Self-Amending Crypto-Ledger Position Paper" (http://tezos.com/position_paper.pdf).
I would like to thank lead Nxt developer Jean-Luc Picard for conversations via dev maillist, and Bill White
for very thoughtful conversations.