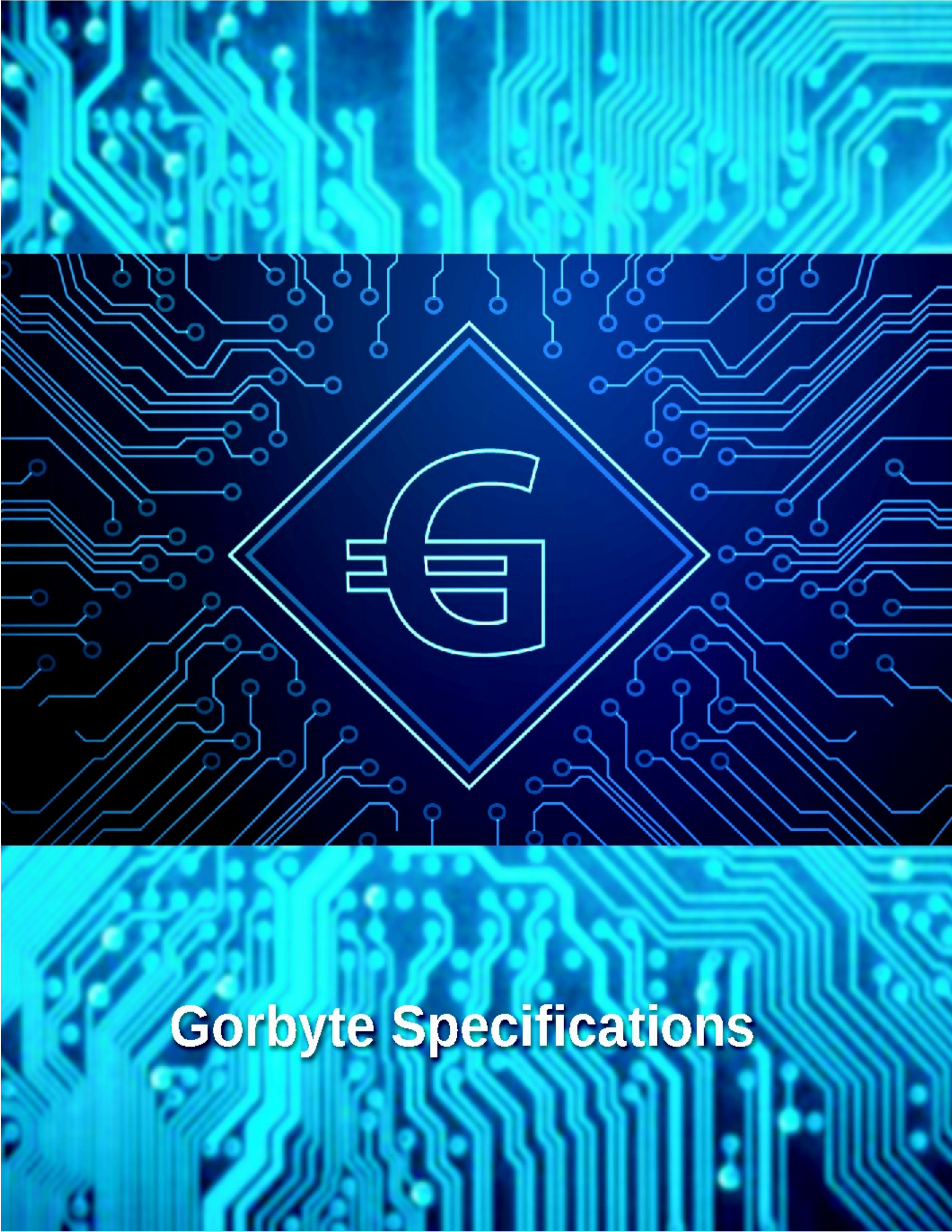

Gorbyte Specifications

Gorbyte Technical Notes

Table of Contents

1. Preface

This document is part of a series of Gorbyte specification documents. These documents will be published at the opportune time.

These Technical Notes support the “Gorbyte Introduction” white paper and provide a few more technical details about Gorbyte and competitive unpermissioned networks. Repetitions between the two documents are avoided, as much as possible.

2. Introduction

The Byzantine Generals problem concerning network fault tolerance has been studied since 1982 by many researchers in academia[1,2,3,4,5,6,7,8]. The development of Bitcoin, almost thirty years later, proved that an original approach was possible and could be successful in creating a working fault tolerant network for financial transactions.

In the meantime, researchers and developers focusing on the Bitcoin design[9] have found practical limitations (e.g., scalability[10] and energy consumption[11]), possible security problems (See the details in section 5.4) and suggested performance improvements (e.g.: GHOST[12]).

In addition, some Wallets and Exchanges that interface with Bitcoin and similar networks have had security problems, with consequent losses of millions of dollars in currency value[13].

Other researchers have proposed and/or patented various cryptography solutions that improve the design by decentralizing the verification process[14,15,16,17.]

Finally, several implementations have replaced, or are planning to replace, proof of work with some variation of proof of stake[18,19].

In this active research and development environment, the Gorbyte project is a corporate initiative, not an academic exercise.

The main objective of the Gorbyte project is to replace Bitcoin’s consensus mechanism and deploy a scalable crypto-network that can provide a free financial transaction service worldwide, with a much higher throughput.

A secondary objective is to introduce additional user services and features (See “Gorbyte Additional Features” document.).

The design algorithms used by Gorbyte will not be patented. As the project develops, the design details will be made public and the code will be released as open source.

2.1 Brief Description of the Bitcoin Crypto-network

Bitcoin, representative of Proof of Work (PoW) crypto-network designs, has been successful in establishing a digital currency and a method of securely exchanging currency among users. Users are identified to the network through an address (a hash of a public key). The following are the basic concepts of Bitcoin[20].

The Bitcoin network relies on a block chain which is a shared public ledger.
A financial transaction represents the transfer of currency to a destination address. All confirmed transactions are included in the block chain. In this way, Bitcoin wallets can calculate their spendable balance and new transactions can be verified to be spending bitcoins that are owned by the spender.

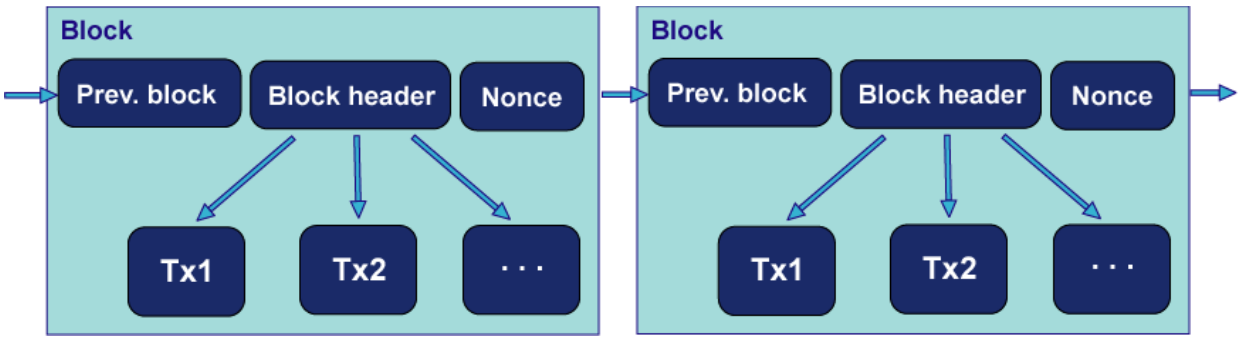

Diagram 1: Blockchain data structure

The integrity and the chronological order of the block chain are enforced with cryptography. A transaction transfer value between Bitcoin wallets. Bitcoin wallets keep a secret piece of data called a private key or seed, which is used to sign transactions, providing a mathematical proof that they have come from the owner of the wallet. The signature also prevents the transaction from being altered by anybody once it has been issued.

All transactions are broadcast between nodes and usually begin to be confirmed by the network in the following 10 minutes, through a process called mining.

Mining is the consensus process that is used to confirm pending transactions by including them in the block chain. It enforces a chronological order in the block chain and allows different computers to agree on the state of the system. To be confirmed, transactions must be included in a block that conforms to very strict cryptographic rules that will be verified by other nodes.

These rules prevent previous blocks from being modified because doing so would invalidate all following blocks. Mining also creates the equivalent of a competitive lottery that allows the random choice of one miner, among those that can demonstrate their processing power, to compose and distribute the current block. This lottery prevents any individual from easily adding new blocks consecutively in the block chain.

2.2 Proof of Stake Alternatives

In those crypto-networks where PoW is substituted by Proof of Stake (PoS), the validation functionality (see definition in section 3) is assigned to a set of validators.

A formal description of a PoS system was published in 2014[21].

There is a conceptual difference between permissioned blockchain-based networks and unpermissioned PoS crypto-networks:

- Permissioned networks rely on a fixed set of authorities that need to be trusted a priori, while
- PoS systems use validators that are usually randomly selected, as well as selected on the basis of their currency holdings. They earn trust by putting their money at stake.

There are several variants of the PoS system. Usually nodes need to subscribe as validators by declaring their currency holdings. These may however change over time, and so does trust in the validators.

The PoS mechanism randomly selects from this set one or more validators for the current block. These validators are then responsible also for distributing the block.

One way to implement PoS is to make the probability of a validator being selected proportional to his currency holdings. If a validator times out, then another one is chosen.

When multiple validators are used, then a traditional consensus algorithm must be used to agree on the composition of the current block among validators.

Other variations to the PoS scheme are not necessarily related to a person's currency holdings. In some proposals, such as ALGORAND (Ref.[14]), any node can potentially be selected as a validator, through a lottery based on a function of the validators key and the last block hash.

The lottery mechanism reduces the number of validators from all nodes to a much smaller set. One of the issues is validating the validators themselves. Prof. Micali, of ALGORAND, proposes a repeated process of random selection of validators, several times for each block.

In general, PoS systems are much less expensive than PoW systems. However, they are not trivial. It is not known how much the various validations and controls may affect the efficiency and security of the network. They have not been subject to as heavy testing as PoW.

One author[22] claims that distributed consensus is not workable with PoS.

A comparison between common PoS designs and Gorbyte is included in Appendix 5.2.

2.3 The Basics

Current crypto-networks propose and implement solutions that improve security and reduce centralized control through encryption. They use encryption and hashing for consensus.

The Gorbyte design relies on encryption for security, for node address verification and uses hashing for correcting and minimizing transmission. However, it introduces its own protocols for establishing a random logical network of participating nodes, for reaching consensus among them, and for block equalization. The limits demonstrated by the theoretical work on the Byzantine Agreement problem and the limits demonstrated by the CAP theorem apply to all distributed system designs.

The Gorbyte crypto-network is characterized by the decentralization of two key logical components of crypto-networks' design: the consensus process, and the responsibility for identifying attacks based on multiple node identities or multiple transactions (DoS).

- The Gorbyte consensus process includes the protocols used for reaching a majority agreement about the composition of a block, and to guarantee that the exact same copy of the block is added to the blockchain on each node.
- In Gorbyte, the responsibility for early identification of DoS and majority attacks is decentralized through the functionality of the BRUD device, a virtual or physical ***b****lockchain* ***r****egistered* ***u****nique* ***d****evice*.

The BRUD procedures, which involve the BRUD device registration on the blockchain and the verification of the BRUD device uniqueness, have no relation with the authorization of transactions. Transactions are visible to all, but are not authorized or controlled by any entity group, or third party.

The Gorbyte crypto-network is unpermissioned.

3. Individual Crypto-network Components

As it was analyzed by Richard G. Brown[23], the functionality of a crypto-network can be categorized according to the following functions, classified by components:

Consensus
The ability to reach consensus on the exact content of the ledger across all nodes in a decentralized, trust-less network, where each node contains a replica of the ledger. This problem, has been theoretically described, for fault-tolerant networks; it is referred as the Byzantine Agreement problem. Consensus must be achieved even if a number of peers in the process cannot or will not participate, or will not play fairly.

In Gorbyte the concept of consensus has changed from the Bitcoin randomly centralized consensus (i.e., one random miner decides about the proposed block composition and its addition to the blockchain), to a consensus mechanism (MARPLE) where most active nodes come to a concurrent agreement on the proposed block composition and its addition to the blockchain.

Authentication
Distributed, private authentication of transactions is achieved through private encryption keys generated by individual users. There is no master key or password allowing an administrator or authority to authenticate or block transactions in the system.

In Gorbyte the concept of authentication has been extended to include the suppliers of BRUD devices. The suppliers initially register a device on the blockchain. BRUD devices are used to verify the uniqueness of their temporary association with a particular node address.

Validation
The functionality that guarantees that each update to the system (each transaction) is operating on the correct ledger, on the correct ledger entry and performs a valid operation, such as moving an amount of currency from an unspent output to a valid address. All nodes must be able to perform validation and must agree on the validation rules.

In Gorbyte all active nodes perform this functionality. However, the concept of validation has been extended to include the validation of new sessions a node tries to establish, through the node/BRUD device association. This helps limit the number of nodes a user can control, buy or simulate. While this limitation has little impact on user functionality, it allows Gorbyte to include limitations on the number of addresses per user, the number of sessions per node, and the number of transactions per address, per block.

Uniqueness
In addition to the validity requirement, each transaction must be unique (e.g.: the anti double-spend requirement). Two valid transactions, generated by the same or by different players, cannot conflict with each other. If a conflict is detected, then a resolution must be found whereby only one of the transactions is chosen and recorded.

In Gorbyte the concept of uniqueness has been extended to include the already mentioned BRUD functionality, which verifies the "Node Address to Device Association" (NADA). The uniqueness of this association allows Gorbyte to introduce limitations protecting the network from Denial of Service and majority attacks.

Immutability
The mechanisms by which the ledger is maintained and verified in each node, in such a way that, if a modified copy is used by any adulterous nodes, all other nodes will be able to detect the change. Thus the immutability of the replicas of the ledger is guaranteed in all unadulterated nodes. As Brown puts it: "once a transaction is committed, nobody else will accept a transaction from me if it tries to build on a modified version of some data that has already been accepted by other stakeholders".

In Gorbyte the concept of immutability is reinforced by the absence of forks. Forks create a temporary indecision about whether a block is immutable or is going to be replaced. In Gorbyte blocks become immutable immediately, at the end of each block reconciliation period.

3.1 Requirements Arising from Distribution

Given the evolution of the generic bank ledger from traditional, to shared, to global, to replicated (See the taxonomy diagram by Richard Gendal Brown[24]), the following requirements arise, also identified by Brown:

In order to avoid unauthorized changes, i.e.: achieve **immutability**, the ledger is replicated in all network nodes. Hence the requirement for achieving a **consensus** on the exact content of the ledger.

[It is expected that technology will exponentially increase its ability to store volumes of information. This increase will be faster than the increase in blockchain space requirements. For example, the Gorbyte blockchain by 2025 will require less storage than currently used for the Bitcoin blockchain(about 120GB)].

Because of the peer-to-peer nature of unpermissioned crypto-networks, transactions need to be **authenticated** through private keys by the individual end-nodes, and not by a central authority, nor by delegated, trusted intermediaries.

Finally, each node must be able to **validate** each transaction, and exclude any possible **conflict** between transactions.

Gorbyte differs from existing crypto-networks in fulfilling the above requirements.

3.2 The Gorbyte Difference

Current crypto-networks guarantee the immutability of the ledger against malicious attacks and majority attacks through their PoW (or PoS) and rewards to miners or validators.

Gorbyte achieves the same objectives through a cooperative majority agreement process. A node cannot propose a block composition to other nodes without fully participating in the various protocols that are part of the Gorbyte cooperative consensus mechanism.

In addition, the malicious attacks based on multiple transactions from one node, or majority attacks based on proliferation of identities are avoided thanks to the **b**lockchain-**r**egistered **u**nique **d**evice (BRUD). Its node address to device association is guaranteed by the procedure for registration of the BRUD device on the blockchain by the device suppliers, and by the verification of the uniqueness of device performed by peer G-nodes.

BRUD devices will become part of a Gorbyte node, much the same as other integrated accessory devices. Initially BRUD devices can be virtual, but they will evolve to include different models, as technical progress is made in biometric research, until the device will include most of the functionality of a mobile communication device, a tamper-proof blockchain wallet, and a crypto-network node[25].

Initially, Gorbyte will use both its cooperative process and its BRUD devices to reinforce its defenses against multiple identity attacks and Denial of Service attacks.

Gorbyte's design allows for providing a free financial transaction service and a free lightweight distributed processing service. A fee will be required only for more complex distributed processing applications.

The other components of Gorbyte, at the transaction level, are the same or equivalent to the corresponding components in Bitcoin, in order to exploit its track record and to allow for compatibility with existing sidechains, exchanges and wallets.

The major functional improvements introduced in Gorbyte, in addition to its new consensus mechanism, are described in a separate document (See "Gorbyte Additional Features" document).

3.3 Why Gorbyte Uses a Cooperative Consensus

Gorbyte can use a cooperative consensus process because it includes other defenses (i.e. BRUD devices), in addition to its distributed PoW against Denial of Service and majority attacks.

The following are the main reasons why Gorbyte uses a cooperative consensus process among all active nodes, rather than a competitive process:

- It resolves conflicts synchronously, during its cooperative consensus process. Thus, no conflict blocks are created and forked, resulting in a cleaner design.
- It is a simpler approach, from the architectural point of view: All active nodes work concurrently and recursively, in cooperation with nei-peers, within their logical environs, until a common agreement on the block composition is reached. Differences are then handled efficiently.
- Any internet connected device (over minimum requirements) can participate as a node (G-node). There are no appointed, selected
or special nodes in charge of validation and block composition/distribution.
- It reduces processing time so much that download speeds, not processing time, become the throughput limiting factor. Thus Gorbyte's throughput is three orders of magnitude higher.
- It eliminates some of the possible security problems, such as the verifier's dilemma (explained in section 4.1.5), or stake grinding[A], since there is no advantage for one node to be faster than others, or be selected as a validator.
- It eliminates rewards to miners or validators, and fees for basic transactions, reducing its operational costs to essentially zero. Thus, Gorbyte can sustain a business model providing no-cost basic transactions, and sharing the profits obtained from more complex general distributed applications (GApps) with currency holders.
- It allows for the elimination of miners and their rewards. As a consequence, special mining processors used in PoW solutions are avoided. Thus Gorbyte's design reduces the global energy consumption, with respect to PoW designs, by many Megawatts per day. (See: "How Can Gorbyte Reduce Energy Consumption").
- It allows for faster and definitive confirmations of transactions.

(A) In Proof of Stake systems an attacker could change parameters in its favor of becoming a validator.

3.4 Architectural Layers

The TCP/IP stack has provided a workable solution for many years and new functionality has been added to the stack in such a way that it would work, within the original design.

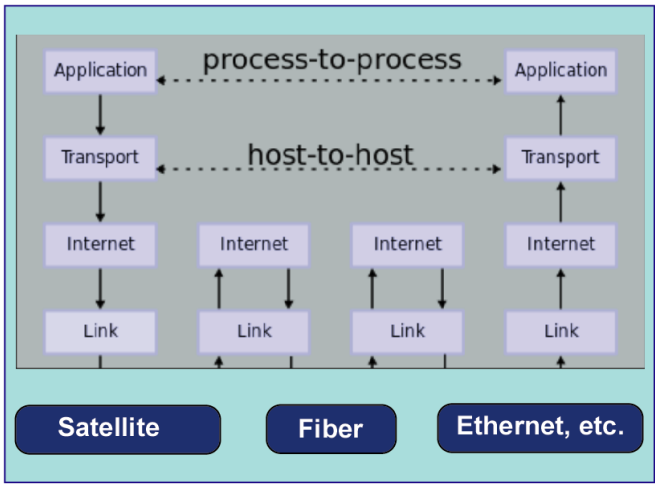

Diagram 2: TCP/IP stack

We can show how Gorbyte fits with the existing TCP/IP structure as follows:

7 Distributed Application Layer	Future **General Distributed Applications (GApps)**.
6 Distributed Operating Environment (DOE)	Scripts for financial transactions. **Graphical API.** **Integral Gorbyte Services**. **Fundamental Blocks** (**C,A,R,S**)
5 Session Layer (A random network of Gorbyte nodes)	In-session messages, Transaction exchanges, script language support, transaction verification. Blockchain support. MARPLE protocol providing the consensus mechanism in a **random network** configuration.
4 Session Establishment and Verification Protocols	Automatic Multi-Session Latching, node initialization, BRUD protocols, SFI mechanism.
3 IP Transport Layer	TCP/IP Stack, UDP/IP stack (sometimes referred to as Network layer + Transport layer), or other.
2 Link Layer	Any device-to device, node to node protocol.
1 Physical Layer	Ethernet, Fiber, Satellite, etc.

Table 1: Mapping the communication layers with the Gorbyte functionality

3.5 Compatibility

Maintaining the same security structure for addresses and transactions, allows Gorbyte to be compatible with third party hierarchical deterministic standard wallets, payment protocols, currency exchanges and sidechains.

In the longer term, the Distribute Operating Environment will allow compatibility among general distributed applications (GApps), as far as communication, authorization, data replication, and security. This will prompt the easier development of existing and new applications and devices based on the blockchain.

4. Bitcoin: Not the Solution of the Future

Bitcoin has advanced the concept of unpermissioned, public crypto-networks and the concept of using a blockchain for financial transactions. It has been successful as it was the best solution at the time. It works, but it uses a large amount of real energy for mining.

One of the shortfalls of Bitcoin is that a significant portion of its mining functionality (the hashing functionality required for PoW) has been implemented in specialized hardware (ASIC). This has caused users with normal devices connected to the internet to be excluded from the mining process.

As the difficulty level raises, and mining becomes more expensive, the Bitcoin hash race makes less and less sense, both from the economic and the technical points of view.

Gorbyte restores the concept that any node should be able to use free financial services by participating as a network node. It achieves this by using the processing resources of each active node for useful, expanding functionality.

4.1 The Economics of the Bitcoin Network

The cost of specialized hashing processors and the cost of electricity used in running those processors amounts to more than $1M per day, with today's network size.

This cost is eventually paid by Bitcoin users.

4.1.1 Technical Aspects of Bitcoin

After the introduction of Bitcoin, researchers found security problems and proposed related solutions[26]. Some of the solutions have been implemented. Other solutions are more difficult to apply.

Even so, Bitcoin has managed to become very successful as far as actual use.

4.1.2 Proof-of-Work and Forks

The reason crypto-networks use the mechanism of a hash race is not related to maintaining the integrity of the block chain, but is related to the problem of selecting and propagating one block and making sure that the exact same block is stored by all nodes in the blockchain, i.e. the replication of an exact copy of each block. When more than one node is involved in the competition (in PoW systems), or in the selection of validators (in PoS systems) the mechanism of forking is used to resolve conflicts.

The Proof of Stake (PoS) approach and its comparison with Gorbyte will be discussed in section 2.2.

The bitcoin solution is not the best[A,27] solution for achieving agreement on a common pattern, within a set of participants. There are also other technical problems inherent in the forking mechanism[28].

Using the *shortest block header hash* for PoW encourages a race to more and more hashing power to keep up with other miners. After a number of months such specialized processors become obsolete.

Using this PoW approach is very expensive, as it is performed in parallel by all miners.

(A) "A primary consideration regarding the operation of blockchain protocols that are based on proof of work (PoW)—such as bitcoin—is the energy that is required for executing the protocol. At the time of this writing, generating a single block in the bitcoin blockchain requires a number of hashing operations exceeding 2^{60}, which means that significant energy needs to be expended in order for the protocol to run. Early calculations placed the energy requirements of the protocol in the order of magnitude of a country. This state of affairs has motivated the investigation of alternative blockchain protocols that would obviate the need for proof of work by substituting it with another, more energy efficient, mechanism that can provide similar guarantees. It is important to point out that the proof of work mechanism of bitcoin facilitates a type of randomized 'leader election' process that elects one of the miners to issue the next block."

Architecturally, this is a centralized solution with a random center for each block cycle, not a truly distributed one.

4.1.3 Unconfirmed Transactions in the Mempools

Typically, all Bitcoin transactions are transmitted from node-to-node over a peer-to-peer network, and are stored in mempools (i.e., the record of unconfirmed transactions) of individual nodes. When a miner finds a block (i.e. wins the hash race), it may have included some of these transactions in that block. Subsequently it transmits this block over the same peer-to-peer network. Of course, this means that all transactions in the block are effectively sent over the network twice: once as a transaction, and once as part of a block.

Miners may not fill a block, but still maintain some transactions in mempools. This creates a lag between the time transactions are issued and blocks are confirmed.

The above problems do not exist with Gorbyte. In Gorbyte, the composition of the current block completes synchronously. After a consensus has been reached, Gorbyte uses the THLR technique (*Transaction Hash List and Replacement*), during the equalization process, to reduce double transmission of specific transactions to exceptional cases.

4.1.4 Scalability

Today the Bitcoin network is restricted to a sustained aggregate rate of approximately 7 tps (transactions per second) due to the Bitcoin protocol restricting block sizes. At the same time, it is having bandwidth problems, thereby not allowing a faster transaction rate.

This is causing considerable opportunity for backlogs in the network[A].

To increase the transaction rate, several initiatives have been proposed and adopted to reduce the response times of nodes, the traffic volumes the size of messages, and the time spent in mining.

Another aspect of scalability is the ability to add functionality. Bitcoin, for example, would require additional functions in order to use the network as the basis for distributed applications.

Soft forks are normally used to make changes or add functionality to Bitcoin, but these require a considerable amount of coordination[B].

When changes are impractical with soft forks, sidechains[C] have been proposed and designed, such as Blockstream.

Whether such solutions are practical and secure or not, they demonstrate the need for functional scalability in Bitcoin.

The Ethereum crypto-network has been created to solve some of these problems.

Ethereum is using the GHOST technique to reduce the time required to compose a block and the confirmation time.

Ethereum uses their Solidity smart contract language[D] to allow the network to handle transactions of any type.

However, Ethereum still uses the same basic mining technique as Bitcoin, and the problems mentioned above are only delayed[E]. For the above reasons, Ethereum has announced their goal to move from PoW to PoS.

(A) See: https://www.cryptocoinsnews.com/almost-1-billion-worth-bitcoins-stuck-transaction-backlog/

(B) See, for example: https://bitcoincore.org/en/2016/10/27/segwit-upgrade-guide/

(C) See a sidechain explanation at: https://gendal.me/2014/10/26/a-simple-explanation-of-bitcoin-sidechains/

(D) The Ethereum use of a Solidity language for financial transactions and smart contracts is perceived by many researchers to be a problem. Gorbyte will use a script language for financial transactions and Distributed Objects for distributed applications.

(E) Cf. CPU, Storage and communication bottlenecks at: https://en.bitcoin.it/wiki/Scalability

4.1.5 The Verifier's Dilemma

Researchers at the National University of Singapore have identified several issues that are characterized as *The Verifier's Dilemma*[29].

The verifier's dilemma arises in any crypto-network that has a high block verification cost. They show that when a script execution requires nontrivial computation effort, practical attacks exist which either lead miners to accept incorrect script results or waste miners' computational resources.

Nakamoto consensus permits unverified blocks

The researchers show that miners lack immediate economic incentives to verify transactions in newly mined blocks. The computational effort exhausted in verifying transactions detracts from the race to mine subsequent blocks, and it is possible that these pools skipped verification in order to gain computational advantage in the mining race.

Essentially, the rewards of the hash race provide an incentive for miners to accept unvalidated blocks.

In Gorbyte, such a negative incentive does not exist[A]. On the contrary, in Gorbyte's consensus process there is an incentive for correctness.

Thousands of decisions happen in parallel and recursively, each node comparing results from nei-peers. Each of these decisions gives a preference to the majority proposals of each of the logical environs.

Resource-exhaustion attack

The same researchers show how a potentially malicious miner does not lose anything by including resource-intensive transactions[B] to his own block, while other miners have to spend a significant amount of time verifying those transactions. Consequently, the attack not only exhausts other miners' resource, but also gives the attacker some time ahead of other miners in the race for the next block.

With Gorbyte, there is no advantage to being computationally faster than any other node. There is no race going on, and there are no privileges to be won.

A Gorbyte node can only issue about a transaction per minute[C]. If an attacker issued transactions that are intended to waste resources, they can only do so once per minute.

(A) If the proposed block composition of one node is substantially different from the one proposed by its logical environ, then this node's proposal is disqualified and the node must adopt the block composition proposed by the majority of its nei-peers.

(B) From the same source: "One can create a block-size transaction which requires miners to hash 19.1 GB of data and takes an average of CPU 3 minutes to verify. Bitcoin patched this vulnerability by allowing only pre-defined standard transactions, and thus limiting the potential applications of Bitcoin. Ethereum, on the other hand, has no such restrictions and permits users to define arbitrarily contracts."

(C) This is possible because of the limitations imposed by BRUD devices.

Since complex transactions in Gorbyte are charged a fee[A], the attacker would spend resources without obtaining any gain.

With Gorbyte there is no advantage gained from skipping the relatively simple computations required for its cooperative consensus algorithms, while there are slight disadvantages for **not** participating:

- the node will not be able to issue transactions for a few minutes; and
- the node will have to re-synchronize, thus use more bandwidth and processing power.

As a result, Gorbyte can increase the flexibility of its allowed transactions without creating an incentive for this type of attack.

The researchers conclude: “We consider it an interesting open problem to determine whether one can incentivize robust computations to execute correctly on a consensus computer *by modifying its underlying consensus mechanism*.”

Gorbyte’s design has done what these researchers hinted, by including a mechanism for cooperative consensus by majority agreement.

(A) Only basic transactions are free. More complex transactions are charged a fee.

5. Comparison tables

5.1 Bitcoin Consensus Functionality Comparison

The following table compares Bitcoin with Gorbyte, with respect to the consensus functionality and its implications:

Bitcoin	Gorbyte
The cost of PoW (the hash race) is high, for the special processors and the energy needed to run them. Thus, high rewards to miners must be maintained. Currently the cost is over $1M/day. Thus, it may attract malicious attackers.	There are no mining rewards. The cost of network security and maintenance is only the cost of the voluntary resources, paid for and shared by each user. This is a distributed and scalable Proof of Work.
The cost of PoW tends to inflate the currency by about 4% per year. The Bitcoin value raises as per demand and goes down as per inflation (paid rewards and fees).	No inherent inflationary mechanism.
Contrary to the philosophy of crypto-networks, Bitcoin's PoW is randomly centralized: One winning miner composes and distributes each Block.	Its Proof of Work is completely decentralized.
Bitcoin's consensus is not cooperative, but competitive (the miners' hash race).	Cooperative consensus by majority agreement among all active nodes.
Blocks already processed may be changed (forked) after being added to the blockchain, thus confirmations are delayed.	Transaction confirmations can be issued immediately after the current block is added to the blockchain. No mechanism exists to modify blocks already stored.

5.2 Proof of Stake Designs Comparison

Looking at the list of issues below, common to PoS designs, we can see how most of these depend on the initial design choice of having a selected number of validators.

Proof of Stake	Gorbyte
Transactions are verified by a selected group of validators.	Transactions are verified by individual users and are not authorized by any centralized entity or selected group.
PoS system may require a high level of locked up investment, to attract validators. This can be good for currency demand, but it is also a cost.	Gorbyte has no investment requirements, although its users may hold currency.
Validators need to be rewarded. This is an operational cost to the network.	Gorbyte operational network costs are zero.
A mechanism may be needed to economically penalize the validators when they approve an invalid or equivocal block. Rules may need to be imposed for them not to leave the scene (i.e., cash their coins) for a certain time.	No equivalent problem can arise in Gorbyte.
In most implementations with multiple validators forks are still possible, as two validators can propose a different block at about the same time. Forks, and the related problems, need to be resolved.	Gorbyte has no mechanism that can generate forks.
PoS crypto-networks are still subject to majority attacks. Those attacks may not be detected until they have already happened. A mechanism is then needed to revert to a valid fork.	Gorbyte uses BRUD devices for early prevention of majority attacks.
Most PoS systems require a transaction fee to avoid DoS types of attacks. Thus their services may not be affordable to many or may not be competitive with a free transaction service.	Gorbyte used BRUD devices for early prevention of DoS attacks. There are no fees for financial transactions.
Every time a restricted number of nodes has the power to validate a transaction, this power can be used to exclude a transaction. Thus, anti-censorship controls have to be devised. Several alternatives exist.	In Gorbyte all active nodes participate in its consensus mechanism.
As validators are in charge of block distribution, the problem of double transmission of every transaction, mentioned for Bitcoin, remains unsolved.	Gorbyte avoids the double transmission problem through its THLR technique.
A class of attacks is based on the idea that the attacker could change parameters in its favor of becoming a validator (stake grinding). Measures need to be devised against this class of attacks.	This problem cannot arise in Gorbyte.

5.3 Communication Functionality Comparison

The following table compares Gorbyte's communication functionality to Bitcoin's transaction level communication. Gorbyte defines a random network on top of the internet requiring sessions to be established and verified among nodes, before they can exchange transactions and participate in the consensus process.

Bitcoin	Gorbyte
Transactions are subjectively selected and included in blocks without order.	Transactions in each block are canonically ordered, and verified, before blocks are reconciled.
Blocks are asynchronously composed by each miner.	Block composition and reconciliation is synchronous.
The current block (1MB of information) is broadcast by one miner to all others. This takes more time as the network grows.	Most G-nodes can come up with a valid equivalent assembled Block. Then a concurrent mechanism is used to reach agreement by communicating between logical nei-peer nodes that are physically randomly dispersed.
No mechanism to detect or limit the acquisition or simulation of multiple nodes by a single party, until the party has 51% ownership of the total network processing power.	The responsibility for preventing the proliferation of identities is distributed. This allows for the verification of the NADA (Node address to Device Association) through the BRUD device registration on the blockchain and its verification of uniqueness.
There are fees for all transactions. A reason fees are needed is to prevent DoS attacks.	There are no fees for basic transactions. DoS attacks are discouraged by limitations on the number of transactions per node, per block.
No comparable mechanisms to improve scalability.	The AMSL (*Automatic Multi-Session Latching*) process, including the DPD (*Datagram Peer Discovery*) and SFI (*Sharing Fingerprint ID*) mechanisms are included[A]. These protocols guarantee faster sharing of information among peers and reduce the probability of network segregation.
Transactions are first exchanged, then re-transmitted with the winning block, thus essentially transmitted twice.	A Block Equalization mechanism (THLR technique) is used, to make blocks identical and minimize bandwidth requirements.

(A) See Section Error: Reference source not found for details.

5.4 Security and Uniqueness Comparisons

The following tables compare the **incentives and rewards** for a hacker to change or manipulate a block in the blockchain, between Bitcoin and Gorbyte.

Bitcoin	Gorbyte
About $1M/day in miner rewards and transaction fees.	No equivalent incentive for hackers.
The rewards derived from deleting or modifying transactions (e.g.: double spending) are caught at the transaction verification level.	Same as for Bitcoin.

If a hacker wanted to replace **a previous block** in the blockchain, what would it have to do?

Bitcoin	Gorbyte
Broadcasts a fork containing all blocks, from the targeted block and including the current block, such that this fork is the highest in the network and satisfies all of the hashing criteria for it to be accepted. The probability of this, according to calculations and experience, is very small. A new supercomputer, or a pool of computers would have to be faster than the rest of the computers in the world. However pools do exist, controlling up to 30% of the Bitcoin processing power.	The forking mechanism does not exist. There is no mechanism in the GCC code (the reference client implementation) which could substitute or update multiple blocks in the blockchain. If a node added a different block or a sequence of blocks to his blockchain it would be out of synch with the rest of the network, its transactions based on those blocks would not be verified and the node would have to re-initialize.
An attacker would have to introduce enough new processors to achieve over 50% of the processing power of the miners in the network. These processors could then take over the blockchain. They could all be in the same location (i.e.: China).	The first protection against a majority attack is the BRUD device mechanism, which limits the number of node addresses per person or entity. Consequently, parties or people acquiring control of multiple nodes are detected much earlier than when they reach a dangerous level of control. If BRUD devices did not exist, an attacker would still have to introduce enough new nodes to own over 50% all of the PC's in the network. These processors would have to be uniformly distributed around the world.

If a hacker wanted to replace the **current block** in the blockchain, what would it have to do?

Bitcoin	*Gorbyte*
There is no defense against a possible anomaly introduced in a block by a malicious miner, which may not be shared with the network for a time equivalent to several block periods. A single node can broadcast a block satisfying the Merkle root hashing criteria for it to be accepted. This is the **normal Bitcoin operating procedure**, where one miner node dictates its block to the rest of the network. The anomalous block would remain part of the blockchain. The probability of this happening, according to calculations and experience, is small, but cannot be discarded. For this reason, Bitcoin needs to wait for more than one block, preferably six, to be relatively sure that a block is really the chosen one.	A hacker would have to find a way to skew the majority agreement process and dictate a specific block to the rest of the network. If this were to happen, then we would be in the same situation as Bitcoin (described on the left).

APPENDIX: Network and Currency Diagrams

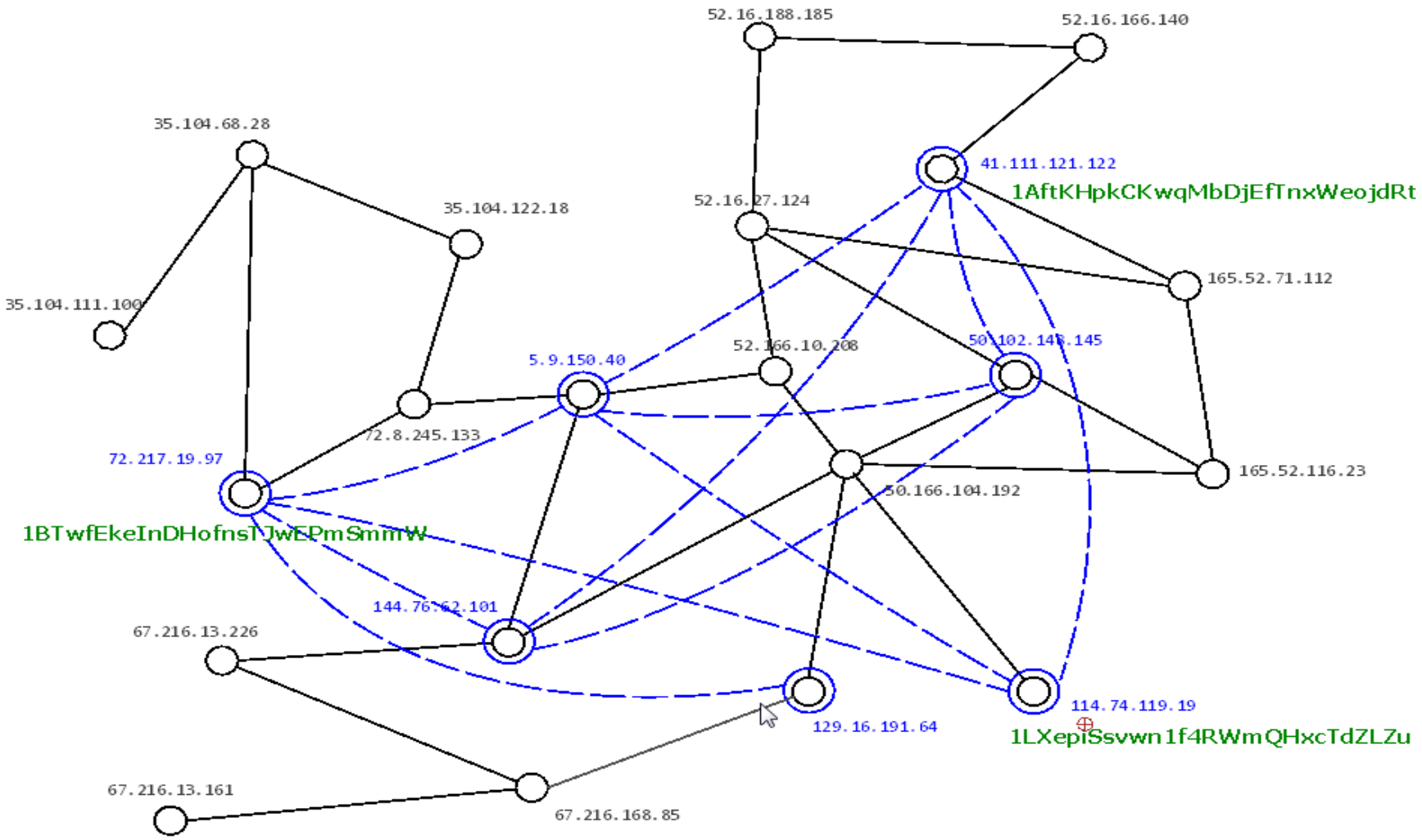

Figure 3: G-nodes over Internet nodes

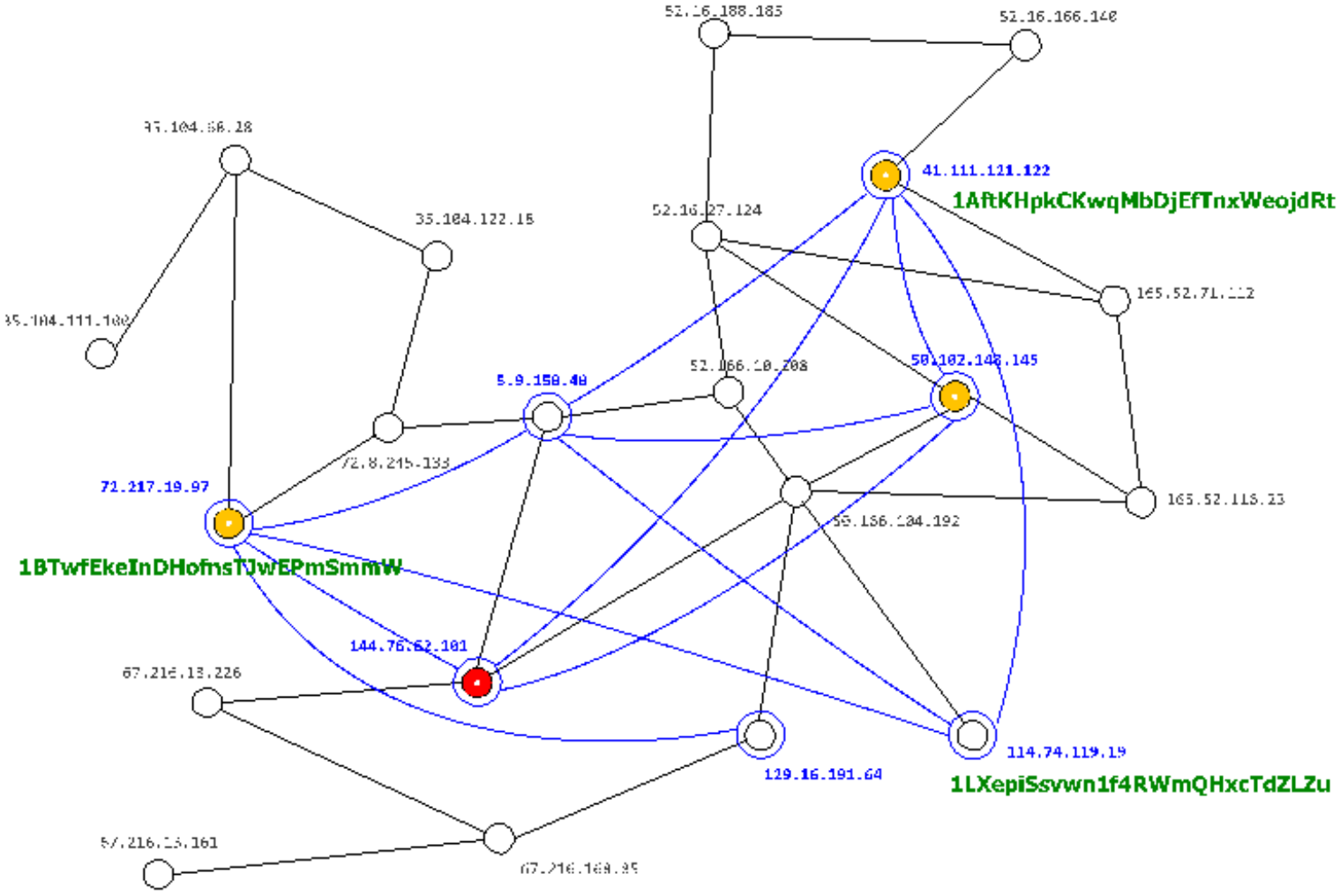

Figure 4: nei-peers of red G-node (orange G-nodes)

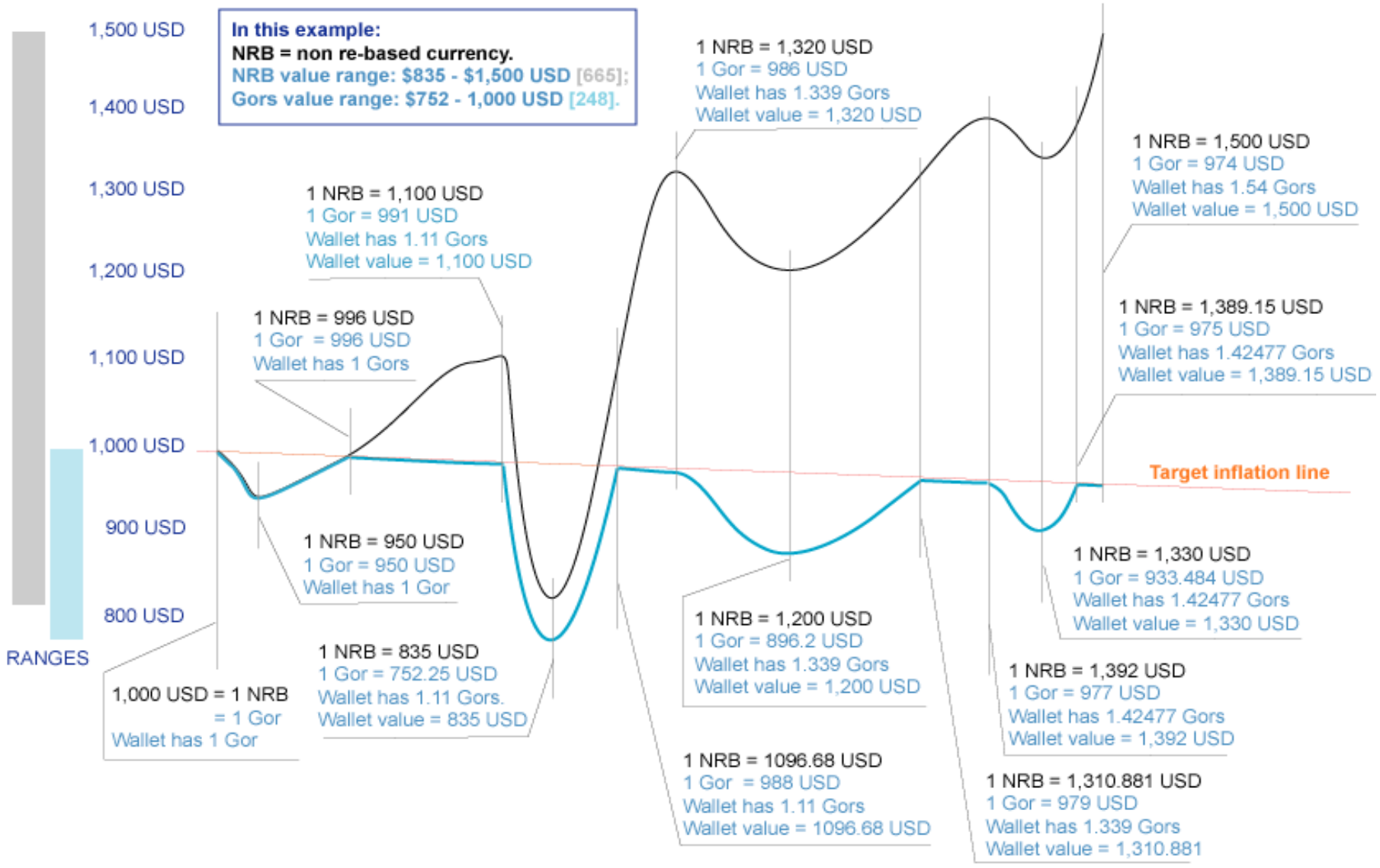

Figure 5: Example of mechanism for currency elasticity

REFERENCES

[1] "The Byzantine Generals Problem", Leslie Lamport, Robert Shostak, and Marshall Pease, *ACM Transactions on Programming Languages and Systems,* Vol. 4, No. 3, July 1982.

[2] Hadzilacos, V. and Halpern, J. Y. "Message-Optimal Protocols for Byzantine Agreement", *ACM Symposium on Principles of Distributed Computing*, 1991.

[3] Chandra, T. D., Hadzilacos, V., and Toueg, S. "The Weakest Failure Detector for Solving Consensus", *ACM Symposium on Principles of Distributed Computing*, 1992.

[4] "Authenticated Byzantine Fault Tolerance Without Public-Key Cryptography", Miguel Castro and Barbara Liskov, Laboratory for Computer Science, Massachusetts Institute of Technology, June 1999.

[5] Baldoni, R., Helary, J.-M., Raynal, M., Tangui, L., Apr. 2003. "Consensus in Byzantine asynchronous systems". Journal of Discrete Algorithms.

[6] "Byzantine Fault Tolerant Public Key Authentication in Peer-to-Peer Systems", Vivek Pathaka and Liviu Iftode, Department of Computer Science, Rutgers, the State University of New Jersey, 2005.

[7] Friedman, R., Mostefaoui, A., Raynal, M., Jan. 2005. "Simple and efficient oraclebased consensus protocols for asynchronous Byzantine systems". IEEE Transactions on Dependable and Secure Computing.

[8] Bessani, A. N., Alchieri, E., Correia, M., Fraga, J. S., Apr. 2008. "DepSpace: A Byzantine fault-tolerant coordination service". In: Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems - EuroSys 2008.

[9] "Bitcoin: A Peer-to-Peer Electronic Cash System", "*Satoshi Nakamoto"*, www.bitcoin.org

[10] "On Scaling Decentralized Blockchains" Kyle Croman et al. At: http://fc16.ifca.ai/bitcoin/papers/CDE+16.pdf

[11] "Bitcoin Mining and its Energy Footprint", Karl J. O'Dwyer and David Malone, National University of Ireland Maynooth, 2014.

[12] "Accelerating Bitcoin's Transaction Processing", Yonatan Sompolinsky and Aviv Zohar, The Hebrew University of Jerusalem, Israel at: https://eprint.iacr.org/2013/881.pdf

[13] E.g. the DAO: http://www.coindesk.com/dao-attacked-code-issue-leads-60-million-ether-theft/ and Mt. Gox: https://www.cryptocoinsnews.com/hong-kong-bitcoin-exchange-mycoin-lost-387-million-customers-money/

[14] Democoin: A Publicly Verifiable and Jointly Serviced Cryptocurrency, *Sergey Gorbunov and Silvio Micali, 2015.*

[15] "ALGORAND The Efficient Public Ledger" Silvio Micali CSAIL, MIT, Nov. 2016.

[16] “A scalable verification solution for blockchains”, Jason Teutsch, TrueBit Foundation, Christian Reitwießner, Ethereum Foundation.

[17] Business Information Exchange System with Security, Privacy, and Anonymity, Sead Muftic, Nazri bin Abdullah, and Ioannis Kounelis, Journal of Electrical and C.E., 2016.

[18] See PIVX at: https://pivx.org/

[19] Ethereum is planning a PoS version called Casper. See: https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/

[20] The following description was extracted from: https://bitcoin.org/en/how-it-works

[21] “Cryptocurrencies without proof of work.”, Iddo Bentov, Ariel Gabizon, and Alex Mizrahi, 2014.

[22] “On Stake and Consensus”, Andrew Poelstra, 2015-03-22, at: https://download.wpsoftware.net/bitcoin/pos.pdf

[23] “*How to explain the value of replicated, shared ledgers from first principles*”, Richard Gendal Brown, at: https://gendal.me/2015/04/27/how-to-explain-the-value-of-replicated-shared-ledgers-from-first-principles/

[24] “Distributed Ledger Technology: beyond block chain”, A report by the UK Government Chief Scientific Adviser, Government Office for Science, 2016.

[25] See “The BRUD Architecture” at: http://gorbyte.com/documents/The%20BRUD%20Architecture.pdf

[26] “Majority is not Enough: Bitcoin Mining is Vulnerable”, Ittay Eyal and Emin Gun Sirer, Department of Computer Science, Cornell University, 2014.

[27] “Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol”, Kiayias, Konstantinou, Russell, David, Oliynykov., April 3, 2017, At: https://eprint.iacr.org/2016/889.pdf

[28] “Information Propagation in the Bitcoin Network”, Christian Decker, Roger Wattenhofer, ETH Zurich, Switzerland, 2013. At: https://eprint.iacr.org/2016/889.pdf

[29] “Demystifying Incentives in the Consensus Computer”, Loi Luu, Jason Teutsch, Raghav Kulkarni, Prateek Saxena, National University of Singapore, October 2015.