MetaHash: Protocol Review

ModernToken Team

August 24, 2018

1 Executive Summary

In August 2018 ModernToken has picked MetaHash project for conducting a high-level review of the blockchain protocol (TraceChain), its architecture, and its implementation as of August 15th.

The promotional materials suggested a very high transaction throughput with low transaction processing and state synchronization times, responding to the perceived industry demand for a fast scalable protocol. An impressive team with a strong pre-existing track record in another high-load IT industry was suggestive of a possibility that the project may be successful at its sought claims.

The input that the ModernToken team has taken into consideration included:

1. MetaHash Yellow Paper.
2. MetaHash White Paper.
3. A personal interview with Gleb Nikitin, MetaHash CRO.
4. MetaGate, the open-source wallet and light client for MetaHash.

The present report contains an overview of the principal design decisions comprising the protocol and their consequences for the characteristics of the resulting system. Overall, the protocol appears innovative in a number of design choices and offers a considerable improvement in scalability, transaction throughput and synchronization times over other known blockchain protocols, while retaining a number of important security properties.

The scope of this study did not include independent replication of the high-load testing, however, ModernToken has found the test results presented in the Yellow Paper to be theoretically consistent with the protocol outline as presented in the provided materials.

2 State of the industry

In 2017–2018 one of the bigger challenges for public blockchains has been transaction throughput under safety assumptions and security guarantees as displayed by Bitcoin network in its early, under-researched years, but with transaction processing time below 5–15 seconds.

Certain problems have been discovered with Bitcoin and Ethereum, and addressed by other blockchain protocols:

1. **Throughput.** The benchmark discussed in the community has usually been that of VISA, namely, practical limits of 5k tps over extended time periods (actually experienced by VISA) and maximal capacity for up to 50k tps (built for VISA and proven by internal load tests). Vitalik Buterin has set this mark for Ethereum in Summer 2017 for the updates of Autumn 2018, and the similar metric is usually kept in mind in relation to Bitcoin-related solutions (see below).

2. **Transaction processing time.** This point has been mostly overcome in most of the existing protocols, with Bitcoin setting the anti-example of 10 minutes, but other protocols having much lower times, about 15s for Ethereum and under 5s for all protocols with PoS, DPoS or BFT consensus. For Bitcoin, one of the proposed and actively developed solutions is Lightning Network (LN), a second-layer protocol over Bitcoin that features near-instant cheap transactions by the counterparties and intermediaries exchanging series of signed Bitcoin transactions that are almost never actually broadcasted to the Bitcoin network itself.

3. **Declining decentralization.** Many blockchains tend to have centralization bottlenecks or potential weak points that allow a single party or a very small number of colluding agents to threaten network stability or affect its function in a significantly disruptive way.[1] This is a standard concern for any DPoS blockchain, based on both theory and practice, but coincidentally similar situations have emerged around Bitcoin and Ethereum at various points in time, whereby a small collection of mining pool owned enough hashing power to be able to break the system.[2]

4. **Centralization of supporting layer.** This is the problem specifically of Bitcoin and Ethereum, inherited by most other public blockchain protocols. As network history growths, the node that are supposed to store it and provide the current blockchain state to the network users are pressed for hardware requirements in storage space, and new nodes need to have both enough space to store the state and enough network bandwidth to catch up with the current events. Unlimited growth gradually prices out consumer-level machines from being able to participate as nodes.

   Since holding nodes is not rewarded in any way in Bitcoin and Ethereum, the only incentive to keep up with costs for node holders is their genuine or financial interest in supporting the ecosystem, i.e. investment presence that requires a good set of nodes to support its function. Therefore the node community is gradually reduced to the theoretical limit whereby the only full node holders may boil down to just a number of organizations with sufficient financial stake in the ecosystem. Which then allows collusion and history attacks as soon as no other nodes remain in the system.

A huge number of proposals, suggestions and approaches have been taken and are being taken by the developer community and various blockchain start-ups to alleviate some or all of these problems.

Bitcoin community mostly ignores problems 4, expects 3 to self-regulate eventually (not entirely without reason), and for 1 and 2 the solutions with traction are either considering Bitcoin "a store of value" that does not need 1 and 2 either, or building second-layer products that do fast and bulk processing outside the Bitcoin blockchain while keeping (or not keeping, it depends) the cryptographic guarantees of Bitcoin. In part the function of Bitcoin trading is taken up by centralized exchanges and regulated custody agents, while an actual attempt to solve Bitcoin's scale problem is the Lightning Network. Which in turn may suffer from centralization of supporting layer, as well as some other technical concerns that have been discussed in theory.

Ethereum community is actively developing Plasma (a second, third, etc. layer protocol over Ethereum) and Casper FFG (a scheme for a PoW and PoS mix partially policed by validating clients with consumer-level hardware requirements). Plasma can solve 1 and 2 and rely on infrastructure solutions for 3 and 4 (such as choosing a correct protocol for internal layers), while Casper FFG, if augmented with sharding (another hard problem Ethereum developers have been trying to solve), can solve all 4. Both products, however, are still some time from release and are not yet proven to be fully functional and secure under reasonable assumptions.

The notorious EOS project has been claimed and repeatedly shown by circumstantial evidence to be centralized, both in theory and in practice, which destroys the purpose of a blockchain solution over a non-chain traditional database. 1 and 2 are non-concerns with EOS, however. The same can be said about Ripple.

3 Protocol overview

Within MetaHash project, TraceChain is a blockchain protocol for value transfers with a focus on high transaction throughput, low network synchronization times, and controlled growth of the state size. The protocol utilizes account model and supports value transactions between accounts at the tested rate of 50k tps with a 3s full synchronization time.

The consensus algorithm is a variation of Delegated Proof-of-Stake without deposit slashing, but with penalties in payout sizes and staking capabilities in response to malignant behaviour: the coins delegated to a node voted as misbehaving are un-delegated and frozen for a set period of time (10 days), during which period they cannot be transferred, re-delegated, or used in staking. This mechanism reduces capital efficiency on the staked tokens, therefore acting as a financial penalty for misbehaviour without actual destruction of funds.

[1] For instance, censoring transactions that are not beneficial in some way to the attackers, or spending the same coins multiple times to receive off-chain goods from multiple parties essentially without paying for them.

[2] There was no published evidence of malignant behaviour actually occurring, but such interference would not necessarily be obvious.

3.1 General structure

There are several kinds of nodes in MetaHash, with each particular machine receiving its node kind from a number of dynamically re-evaluated parameters, including network latency to other nodes, CPU, memory and storage space, recent history, and current geographical and role distribution of other active nodes. There are three separate state chains: one for value transactions ("Main Chain"), one for latency tests ("Network Map DataChain"), and one for technical messages related to the network function: fraud reports and votes, node type changes, block validation signatures from the nodes propagating the newly forged blocks, etc ("Technical DataChain"). 1% of the nodes (randomly chosen and regularly rotated) stays in the testing mode and verifies latency and correct behaviour of other randomly selected nodes, with updates propagating to Technical DataChain and Network Map DataChain as necessary so that the existing network graph and security status of each node are as recent as possible.

Node rewards come from a limited pool of pre-allocated coins plus the transaction fees paid by transaction senders. The reward to each particular node depends on the validity of its role (how many nodes of that type have been active over the set period, as compared to the pre-defined constant sweet-spot number, derived empirically before the network launch), geographical relevance (how many nodes of that type are present in a given region as compared to other regions), how fast the node has been, did it show any malignant activity recently, and how many coins are delegated to stake on that node (compared to all effective delegated coins for all nodes). Unlike most existing and proposed protocols, MetaHash rewards nodes of all types required for the full functioning of the system, creating balanced incentives to run nodes of every required kind.

3.2 Characteristic design features

MetaHash project as outlined in the papers differs from the existing and proposed blockchain protocols in a number of design choices, which, together with other principal characteristics, offer novel or refined solutions to some of the technical challenges known at the present state of the industry.

Role separation within a regulated environment. Every vital kind of node is financially incentivized to be present in the network in a certain sweet spot quantity, with a good enough geographic distribution (measured by location and latency graphs), and to perform well. Node layering by role allows for parallelization of tasks, including validating transaction signatures, verifying transaction correctness, forming blocks, propagating information, and cross-validating each kind of data propagation and storage.

Dynamic role readjustment. The strict distinction of the node types produces the possibility for automatically adjusted node specialization according to node location, its hardware capacities, and its network performance. Dynamic role shifting affords high capability for network readjustment and attack or failure mitigation by reassigning new nodes in place of the attacked or disabled ones, supported with low enough synchronization times that enhance respective reaction times.

Interlinked cross-validation. All the data passing through the network is signed by multiple representatives from each node layer along the way, as they perform their function, including transaction submission, transaction inclusion (i.e. processing), and reporting on the blockchain state to the light client software. Presence of signatures and staking requirements ensure that the Schelling point of each action and response is the correct performance of the protocol function, as any detectable malignant activity (including, among other things, transaction censorship and misrepresentation of the blockchain state) is punishable by temporary stake freezing and penalizing future stake efficiency by a respective vote in the Technical DataChain.

Staking requirements for every node type. The system offers greater resistance to Sybil attacks and DDoS attacks (within the protocol rules, if the aim is to overcome dynamic role readjustment and disable a particular role layer entirely) by restricting accepted interaction types between layers and requiring a valid and minimally bounded freezable stake for running a node with particular access rights.

State snapshots and history purging. Every 250G of history MetaHash publishes and verifies a state block that describes the full present state of all the blockchain accounts. Presence of such a block as finalized allows some of the nodes to purge the history behind that block while retaining the future ability to correctly (and consistently with other honest nodes) validate future transactions as compliant or not compliant with the present consensus state. This approach allows rather cheap nodes to bear

relevance in the state integrity (therefore contributing to security of the account history) without being priced out, so that the supporting layer in its entirety can stay decentralized.

Neither of these design choices single-handed removes a problem or a set of problem outlined in section 2, However, in concert they set up a system of checks and balances that protects the DPoS architecture from a number of potential flaws and problems, such as centralization of the supporting layer (rewards for all node types), planned attacks by groups of nodes (role readjustment, cross-validation), Sybil attacks on infrastructure (staking requirements for every role layer), a subset of DDoS attacks (role separation).

4 Evaluation

MetaHash presents a well designed and thoroughly tested blockchain protocol in its later phases of development. A number of employed mechanics distinguish the project from its conceptual competitors, offering considerable claimed and (internally) tested improvements in vital performance characteristics. The defining points of a DPoS protocol, namely, node interaction for synchronization and block generation tasks, node rewards and penalties, consensus algorithm, and network mapping and adjustment routines, are well described and theoretically consistent with the stated performance expectations.

Claim	**Assessment**
50k tps, as shown by internal load tests.	Likely: shown by MetaHash self-tests, consistent with protocol examination.
Under 3s state update synchronization time.	Likely: shown by MetaHash self-tests, consistent with protocol examination.
Full functioning with 70% nodes going off-line.	Possible: shown by MetaHash self-tests, hard to prove from theory.
Resistance to corruption of up to 90% stake.	Possible, but scenario-dependent, not proven by MetaHash at any point.
Protected by 5 different consensus algorithms.	Factually incorrect, but also irrelevant.

4.1 Consensus fault-tolerance

The consensus problem statement in a blockchain system is two-fold:

1. How to achieve agreement on the current state between the majority of the system participants.
2. Given mechanisms built for 1, how to achieve finality on particular state snapshots, with strong consensus confidence in their validity and historical accuracy.

The benchmark numbers given by theory are 51% (more precisely, $50\% + \epsilon$, for Proof-of-Work chains) and $2/3+1$ vote, the Byzantine Fault-Tolerant (BFT) requirement from the classical article on Byzantine Generals Problem published in 1982. The former is applicable to hashing power, while the latter is usually associated with a pre-allocated voter sets (BFT consensus family and permissioned validation), but is also applicable to Proof-of-Stake (PoS) and Delegated Proof-of-Stake (DPoS) systems, whereby voting is weighted in accordance to the system coins staked to support a particular vote (PoS) or voter (DPoS).

Classic Bitcoin as an example of a Proof-of-Work system does not formally distinguish consensus and finalization, as consensus chain can sometimes lose a block or two from the end in favour of a longer chain mined in parallel, if the hashing randomness and network delays coincide. The 51% rule posits that after a short while the majority of the hashing power will achieve a final consensus on the chain history up to a recent point. This is the reason why in sensitive cases (for large sums) before recognizing a Bitcoin transaction as actually having occurred the network users wait for several blocks to be built and accepted after the block containing the transaction in question, as to expect finality on that particular transaction.

A simple PoS system can be broken with $50\% + \epsilon$ stake, similar to PoW: the majority of stake votes for a particular state of the system, which then gets accepted and consequently finalized. A system utilizing chaining signatures along the signal propagation route can in theory boast a much higher failure tolerance, under certain security assumptions (such as full network connectivity and predictable network latency).

MetaHash voting system with block validation bore by most of the layers along the block propagation route can offer greater confidence in finality and correctness. Any node (and, effectively, stake) collusion,

in order to pass a malformed block or commit other protocol-breaking action needs to cover the entirety of the route that a new block takes, because a single honest node is enough to publish the fraud proof, which, by MetaHash design (and in accordance to the last section of the Byzantine Generals article), is co-signed by every other malignant node, thus exposing the collusion to the rest of the network participants.

Accepting fraud proofs and allowing them to invalidate past network state can theoretically ramp up network fault tolerance to higher percentages. A good theoretical account of that is explored in an article by Vitalik Buterin (published independently and after the MetaHash Yellow Paper), giving account to the aforementioned 1982 work.

MetaHash protocol description is suggestive of a higher stake collusion resistance, since any detectable fraud can be reported via Technical DataChain and entail node penalization, rebuilding votes, trust change votes, etc. Behaviour of that penalty system and its theoretical stability are largely dependent on protocol specifics, which are missing from the presented documentation. Moreover, dynamic shifting of roles can mean varying efficiency of stake collusion, depending on the momentary distributions of the attacking stake and nodes.

From general considerations, ModernToken perceives a claim of resistance to 90% stake collusion to only be viable under different wording and with explicit assumptions: it may be possible that with some small (under 20%) honest nodes with stake, randomly selected among the node roles, can successfully split the network into colluding and non-colluding segments, using the internal voting on fraud proof and network rebuilding system, destroying trust parameters of the nodes witnessed in collusion. It is likely that this arbitrarily chosen boundary of 20% can be lowered with sufficient modeling and proofs, however, MetaHash documentation, while making such a claim, lacks a consistent argument in its favour.

In conclusion, ModernToken regards MetaHash consensus as theoretically consistent and having at least industry-grade level of security (barring implementation errors or problems in the unspecified areas of the protocol). Validity of stronger claims is likely, both in theory and from the outline of the protocol specification, but consistent proofs remain to be published.

4.2 Concern points

The node software and protocol specification remain unpublished for the time being and evade examination. In particular, the unclear points are algorithms for fraud proof publication and voting, node role selection and self-adjustment, and latency reporting and testing with production-scale network size. Although the tests results for throughput, network stability under various massive node shutdown scenarios, and state space size changes are published, the statistics for Network Map DataChain and Technical DataChain are missing. A more in-depth protocol review would include examination of these chains, their statistics and behaviour under various attack or failure scenarios.

With requirements for every role being filled by a large enough community of nodes, and dynamic role readjustment, it is not apparent, what are the stability prerequisites and expectations of the system at production scale, with shifting node states and availability. These parameters remain to be discovered during live testing and at some time after the mainnet launch, as recognized by MetaHash and reflected well in the project roadmap. The self-reported tests on testnet show good stats and stable function under high load over extended periods of time,

There was not enough data to evaluate the mechanisms of Trust changes (the TraceChain node rating assessing correctness of its behaviour over time) and its consequences for potential attack costs.

4.3 Risk assessment

In general, potential dangers for DPoS-based protocols within the formal protocol rules include the following: majority attacks, Sybil attacks on infrastructure, hardware-wise centralization risks, capital centralization risks, attacks on history, and the so-called nothing-at-stake problem. MetaHash protocol design offers industry-level or above industry-level mitigation strategies and defense mechanisms for all of them.

Centralization risks are controlled with rewarding all kinds of nodes for network-supporting operations, including consumer-grade hardware and network bandwidth, thus thinning potential incentive structure for capital-heavy participants. Start-up centralization risks in the coin distribution and launch-time node seeding mostly remain a point of trust and careful selection of investors with reasonable expectations for lack of collusion, and some checks are offered by the team to compensate for the initial node centralization, suggesting external state checkpoints to at least add a level of transparency to historical integrity, until the node community grows decentralized enough. Sybil attacks and some of the DoS attacks are contained through layer separation and staking requirements.

Area of interest	**Expected stability**
Throughput & scalability	High for all practical matters, the technological core may be reused in other projects for future applications.
Transaction processing time.	Expected and self-tested as good for all relevant practical applications.
Level of centralization.	Within initial capital distribution parameters, no model incentives for additional centralization. Dynamic role shifting and cross-validation mitigate potential hazards of centralization on par or better than in other existing or proposed solutions.
Centralization of supporting layer.	Not a concern, since every layer essential for system liveliness is properly incentivized.

The MetaHash own analysis of system-wide risks in the Yellow Paper shows a well-rounded approach to the project and willingness to offer steps for better decentralization and transparency for the community. The section includes suggestions for mitigation strategies for possible attempts of developer-side fraud, such censorship and history attacks during the early stages of network operation, which signals genuine interest in protocol independence from the developers, setting the project apart from several well-known examples notorious in the blockchain community.

Resistance to stake majority attack, claimed by the MetaHash team to be as high as withstanding up to 90% majority of malignant stake, is unclear at the point of the present review, and not documented enough to be proven beyond reasonable doubt. It seems plausible that some large-stake attacks can be unreliable to plan for and therefore easily defended automatically by the system via fraud proofs and Schelling point behaviour, as mutual cross-verification of nodes on various layers and dynamic role readjustment introduce a coordination problem for the attacking node and strong checks and balances maintained by other nodes in most layers. However, the numbers of 51% (50% + ϵ voting power, theoretical PoW breaking point), 2/3 + 1 vote (theoretical BFT requirement), and 90% (MetaHash claim) are hard to match to TraceChain and show consistently from the existing protocol specification, because of the node role distribution that will be dynamically shifting at runtime.

5 Conclusion

From the examined materials, ModernToken rates MetaHash protocol as having high technological potential. If accompanied by a thorough security analysis of a live testnet, source code study and interaction models testing, the protocol can be classified as secure. All of these stages are reflected in MetaHash roadmap, which demonstrates that the company recognizes well the level of responsibility bore by developers of an infrastructure blockchcain protocol.

The main challenge is expected outside the scope of protocol design, but rather in building documentation, a set of tools and, eventually, a developer community around the protocol in order to enable production of consumer-level software leveraging the protocol and, possibly, other applications of the technological core of TraceChain.