Fractional Reservation Based Mempool Processing in Blockchains

Stanislav Kruglik
Skolkovo Institute of Science and Technology
121205 Moscow, Russia,
+7-495-280-1481
stanislav.kruglik@skoltech.ru

Sergey Vorobyov
National Research University Higher School of Economics
101000 Moscow, Russia,
+7-495-531-0059
savorobyov@edu.hse.ru

Yash Madhwal
Skolkovo Institute of Science and Technology
121205 Moscow, Russia,
+7-495-280-1481
yash.madhwal@skoltech.ru

Yury Yanovich
Bitfury
123100 Moscow, Russia,
+7-495-477-4477
yury.yanovich@bitfury.com

Abstract— A massive traffic events that impact all the nodes in the distributed system may cause a Denial of Service (DoS). Managing DoS attacks is even harder in peer-to-peer projects because of multiple equal in rights nodes (miners or maintainers) that communicate across the globe to secure the network. In most blockchains, users can send transactions. Moreover, as a systems throughput is limited, the ability to send transactions should be limited in some honest and transparent way. Otherwise, the pool of unconfirmed (pending) transactions mempool could be overloaded, and it may cause DoS. In Bitcoin, users pay a fee for each transaction to address this issue. Steem.io introduced an alternative approach based on the fractional reservation of the blockchain block space. This approach is an adaptation of similar ones from the network routing and banking systems. The block space fractional reservation for blockchains in terms of a score function is introduced in this paper. Authors made a private blockchain project demo on Exonum framework. The score function influences only on mempool processing, and other blockchains can also make use of it.

Keywords— blockchain; fractional reservation; denial of service attack.

I. INTRODUCTION

DDoS, a class of DoS, is a distributed denial of service attack where a large number of bots makes a massive number of requests to a server to bring down a website or stop online service, making them unresponsive in authenticating requests [1], [2]. DDoS attacks have been on the rise and are becoming increasingly technically advanced in their duration, adaptability to issue special attacks and ability to find new targets, breaking their overall record of attacks [3]. With a DDoS attack in effect, the networks downtime severely impacts the whole system, affecting its productivity, causing physical damage and even threatening public safety [4].

In a distributed computer network, if all nodes are impacted by sending volume of traffic larger than its capability to handle, it may result in interruption or suspension of the services. This is the beginning of a DDoS attack. Blockchain is a decentralized database with tamper-resistant log and built-in auditability that took a prominent place in the field of distributed networks [5]. And subsequently they found applications in many areas (state registers, supply-chain management, biomedicine, finance, etc. [6], [7], [8], [9], [10], [11], [12], [13]). Blockchains can be categorized by the level of access to the data for different types of participants on public and private ones with several further subdivisions [14], [15], [16], [17]. Each of these types has its application area and limitations. No matter the type of blockchain, it can be a public network project without any limitations on user participation.

A DDoS attack on a blockchain would imply that a person attempts to utilize all of the networks resources in a way that the miners are unable to commit to or record any unconfirmed transactions from mempool (i.e., flooding the network with correct but useless transactions). If the rate at which transactions arrive at mempool are higher than the throughput rate is another scenario of DDoS attack. In Bitcoin network, 7 transaction per second rate makes it more vulnerable to spam attacks [18]. Since, DoS attacks are inexpensive to carry out and quite disruptive. Participants of

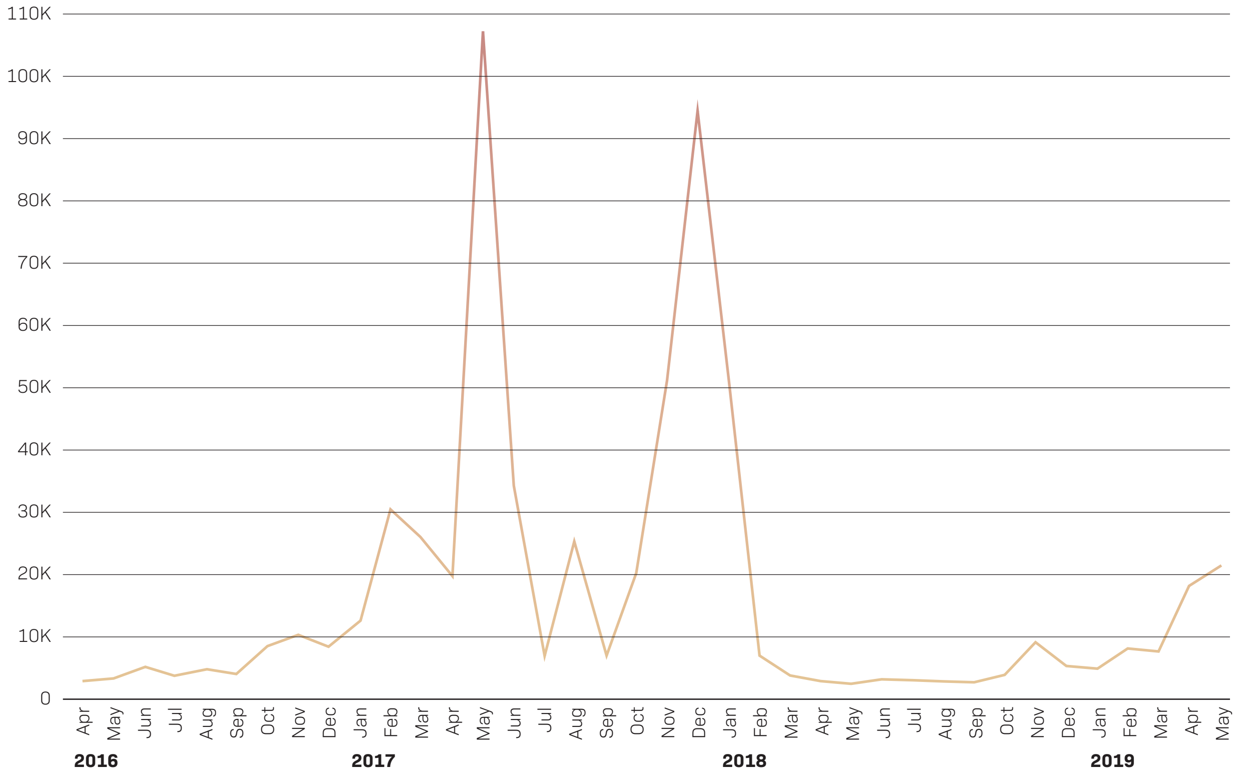

Figure 1. Pending Transaction in Mempool

this network have been common victim of this type of attack [19]. From a business point of view, the addition of pending transactions to the block makes it challenging for users to transact. One of the possible ways to resist DDoS attacks in blockchains is to set a transaction fee [5]. Theoretically, in the case of cryptocurrencies or tokens, transactions with higher fees are more likely to get committed [20]. Miners maximize their income, so they are incentivized to include transactions with a high fee in the next block. As a result, transactions with low fees can await their inclusion into blockchain for years. DoS attacks are inexpensive to carry out and quite disruptive. Participants of any peer to peer network can been an easy victims of this type of attack. Mining pools are easy for DDoS attack [19]. Fig. 1 shows number of pending transactions of bitcoin mempool. In 2017, it was attacked 2 times with sudden increase in pending transactions.

In a private blockchain environment, where an organization maintains its own blockchain, a local Internet Service Provider (ISP) provides a range of throughput to the organization. The cost of private blockchain maintenance is slightly lower than for Proof-of-Work public ones, and there is no need and no motivation except DDoS protection to set significant transactions fees. An alternative approach for DDoS protection is used in network routing and banking systems [21], [22] and it was first introduced for blockchains in Steem.io [23].

In the case of network routing, the ISP has two choices, either to run a full reserve or a fractional reserve. Under a full reserve system, each user is only allowed a fraction of the maximum throughput proportional to their shares. Since not everyone is using the Internet at the same time, the organizations network would be significantly underutilized. Under a fractional reserve system, the individual users could utilize more bandwidth than they are entitled to at any given point in time, as long as no one uses the Internet at the same time. The problem with operating a fractional reserve is that the congestion can occur at any moment when too many people wish to use the network at the same time. Due to it, the ISP needs a way to prioritize the users request during congested periods. In the most extreme case, a fully congested network must revert to a full reserve system. The challenge is in setting the proper fractional reserve ratio.

This paper considers block space fractional reservation for blockchain in terms of score function, i.e., to share throughput among users honestly and transparently. The proposed function definition and implementation are provided. The rest of the paper is organized as follows:

- The score function and intuition beyond it are introduced in Section 2.
- Key points for the score function implementation are presented in Section 3.

- Possible scenarios of DoS attacks on fractional reservation-based system are listed in Section 4.

2. FRACTIONAL RESERVATION IN BLOCKCHAIN

Score function (SF) is used to arrange the queue of unconfirmed transactions during the new block proposal generation. Its goal is to provide an honest block space utilization in case of a large mempool size. The SF is calculated for individual users, taking into account their blockchain throughput usage and balance history.

This paper assumes a tokenized system, where users have wallets to store tokens. The token is assumed to be like traditional money or ERC-20 [24] with interchangeability properties, instead of being associated with Bitcoins unspent transaction output logic. Each transaction has a single input wallet for simplicity purposes.

Note that the proposed approach is also applicable to the blockchains without tokens if the attackers are limited in the ability to generate new users. It is also applicable for state registries, and other systems with strong know your customer (KYC) processes. Users could be assumed to have one artificial unspendable token once they have registered in the blockchain without an internal token.

2.1. User Score Function

Let

- S be the total token supply. For simplicity purposes, we assume that S is fixed and known, i.e., no token emission and burning.
- u be a user's amount of tokens.
- C be the maximum block capacity. In this paper, it is expressed in transactions (tx) unlike Bitcoin and Ethereum, where capacity is measured in terms of bytes (actually, in terms of weight since SegWit [25]) and computational complexity. For example, $C = 2000$ tx.
- T be a given characteristic time window expressed in seconds. For example, $T = 600$, which means 10 minutes.
- L be the number of blocks per T. For example, $L = 600$ blocks per 10 minutes, which means 1 block per second.
- R be a reserve ratio coefficient. Generally, it could be adjusted over time. For simplicity, assume R to be fixed. For example, $R = 1$.
- $M = C{\cdot}L{\cdot}R$ be the number of reserved transactions per T seconds.

S, C, T and L are the blockchain network parameters, which exist independently from the fractional reservation. The discrete-time where 1 "second" corresponds to 1 new block commit into the blockchain, is used.

Let SF of individual users be limited with maximum value $A(u)$:

$$SF_{max} = A(u) = M\frac{u}{S} \qquad (1)$$

Note. The detailed explanation of the fractional reservation in blockchains idea is available in [26] (written in Russian). The article claims to be a translation of some of the Steem white paper versions. However, the exact version couldn't be found, and there is almost no explanation in the current platforms white paper [23]. This drawback is covered by proposing score function and implementing it within Exonum framework [27].

Definition of SF is given in the subsections below, and an intuition is introduced beyond it (see Figure 2). A new user starts with a $SF = 0$, which increases over a time period t with a fill rate equals to $A(u)/T$. The SF is an upper bounded function of time t, where the bound equals $A(u)$. Let T_0 be the time when the SF of a user with u_0 tokens in his wallet reaches its maximum. When his transaction is committed into blockchain at t_1, the SF is reduced by 1, and gradually increased with time to new $SF = A(u_1)$, where u_1 is the amount of his tokens after the transaction. When at t_2, the user receives some tokens, the SF_{max} also increases to $A(u_2)$, which will be attained at time T_2. If no transaction occurred, the user would attain the maximum score function $A(u)$, regardless of the time that has elapsed.

2.2. Score Function When No Transactions

When no transactions occur, SF is an linearly increasing function of t with a slope equal to $A(u)/T$ until it reaches the maximum value $A(u)$.

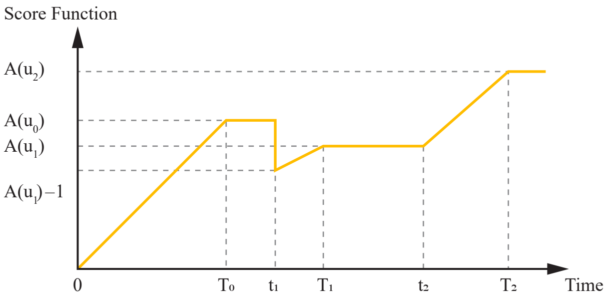

Figure 2. Score Function workflow example

Then SF becomes constant with time when the user neither receives nor sends any tokens. General formula to calculate the SF at time $t_2 > t_1$, when $SF(t_1)$ is given and there are no transactions:

$$\mathrm{SF}(\mathrm{t}_2) = \min\{\mathrm{SF}(\mathrm{t}_1) + \frac{\mathrm{t}_2 - \mathrm{t}_1}{\mathrm{T}} \mathrm{A}(\mathrm{u}), \mathrm{A}(\mathrm{u})\} \qquad (2)$$

Note. The linear growth function is chosen as the simplest possible example. One can use a nonlinear function.

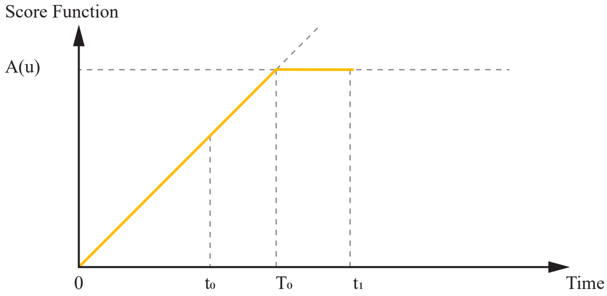

Figure 3. Score Function when no transaction

In Figure 3, one can see the behaviour of score function in the case when $\mathrm{t}_1 > \mathrm{T}_0$ where T_0 is the time when SF reaches its maximum value for u tokens.

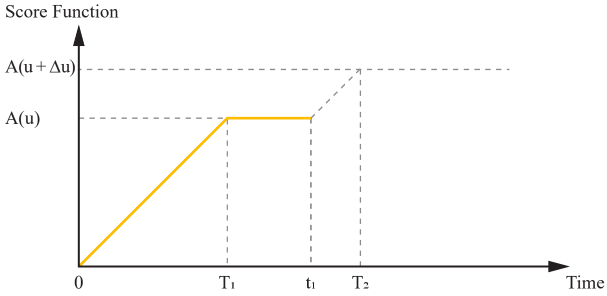

Figure 4. Score Function when user receives tokens

2.3. Score Function for Outgoing Transaction

In Figure 2, when the outgoing transaction takes place at time t_1, i.e., the number of outgoing transactions $\mathrm{N} = 1$, the SF is decreased by 1. The new SF satisfies the following two conditions:

- If a user with $\mathrm{SF}(\mathrm{t}_1) < 1$, sends some token, the new $\mathrm{SF}(\mathrm{t}_1)$ will be 0 as SF is non-negative by design. And therefore, considered as minimum 0 value.
- Else, new SF will be minimum of previous $\mathrm{SF}(\mathrm{t}_1) - 1$ and new maximum $\mathrm{SF}(\mathrm{T}_1)$, i.e. $\mathrm{A}(\mathrm{u}_1)$, where $\mathrm{u}_1 = \mathrm{u}_0 - \Delta\mathrm{u}$, where $\Delta\mathrm{u}$ is the amount of outgoing tokens.

Above criteria are represented by the equation below:

$$\mathrm{SF}(\mathrm{t}\,|\,\mathrm{N} = 1) = \min\{\max\{\mathrm{SF}(\mathrm{t}) - 1, 0\}, \mathrm{A}(\mathrm{u} - \Delta\mathrm{u})\} \qquad (3)$$

If $\mathrm{N} > 1$, transactions from the user are included into a single block, when the equation (3) is applied N times.

2.4. Score Function for Incoming Transaction

In Figure 4, when a user receives $\Delta\mathrm{u}$ tokens from another user at t_1, $\mathrm{SF}_{\max}$ increases according to equation (1)

$$\mathrm{SF}_{\max} = \mathrm{A}(\mathrm{u}_2) \qquad (4)$$

and the slope changes to $\frac{1}{\mathrm{T}}\mathrm{A}(\mathrm{u}_2)$, where $\mathrm{A}(\mathrm{u}_2) = \mathrm{A}(\mathrm{u}_1 + \Delta\mathrm{u})$.

Since then the formulas from subsection 2.2 are in use, i.e., the SF linearly increases from time t_1 till time T_2 when SF reaches maximum and becomes constant until another transaction takes place which will change the SF according to equation (3) or (4).

III. SCORE FUNCTION IMPLEMENTATION

Previously mentioned formulas define SF that is used to calculate unconfirmed transactions priority while a blockchain validator are generating a new block proposal. Each validator computes and stores the score function for all users. SF is not taken into account in block validity check.

Previously mentioned formulas define SF that is used to calculate unconfirmed transactions priority while a blockchain validators, either miners or maintainers, is generating a new block proposal. Each validator computes and stores the score function for all users by itself. SF is not taken into account in block validity check.

For each user, triplet (SF, h, u) is stored, where h is the last height when SF for this users was updated, SF is user's score function at h and u is token balance at h. It is enough to store SF for users with nonzero token balances only. The SF for a given user is recalculated in the following cases:

1. **Pending transaction:** User's transaction is in the pool of unconfirmed transactions.
2. **Incoming transaction:** A transaction which increases the user's balance is committed to the blockchain.
3. **Outgoing transaction:** User's outgoing transaction is committed to the blockchain.

Unconfirmed transactions are prioritized based on SF and first C transactions prioritized by SF descending order are added into the block. A validator creates temporary $\mathrm{SF}_{\mathrm{temp}}$ for each block proposal generation. If a user's transaction is chosen for the block proposal and his $\mathrm{SF}_{\mathrm{temp}}$ is recalculated so that user with highest SF before proposal generation does not get all the space in the block. The block proposal is committed to the blockchain when the $\mathrm{SF}_{\mathrm{temp}}$ is saved as a SF one. Otherwise, the temporary $\mathrm{SF}_{\mathrm{temp}}$ is deleted.

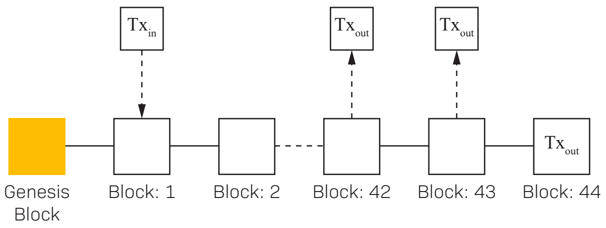

Figure 5. Updating SF on Blockchain

For example (see Figure 5):

- the first incoming transaction Tx_{in} with u tokens to a given user is added to the mempool at height $h = 1$ and it is committed into the blockchain at height 2;
- the validator sets the user's triplet as $(SF, h, u) = (0, 2, u)$ according to the case 2;
- the user sends outgoing transaction Tx_{out} and it reaches the validator before a block with $h = 42$ commit;
- the validator updates the user's triplet according to the rule 1 and Equation (2) and sets $(SF, h, u) = (SF_1, 42, u)$ according to the case 1 and Equation (2);
- the Tx_{out} is not committed into the block 42 and the validator sets $(SF, h, u) = (SF_2, 43, u)$;
- neither the Tx_{out} is committed into the block 43 and the validator sets $(SF, h, u) = (SF_3, 44, u)$;
- the Tx_{out} is committed into the block 44 and the validator sets $(SF, h, u) = (SF_4, 45, u - \Delta u)$ according to the case 3 and Equation (3).

The demo code is available on https://github.com/sergeyvorobuof/fractional-reservation.

IV. DISCUSSION

Although individual users have their score function values, if a user has a huge amount of tokens he can attempt to make a DoS attack on the network, increasing wait times for users with average SF. Two possible attack scenarios are possible:

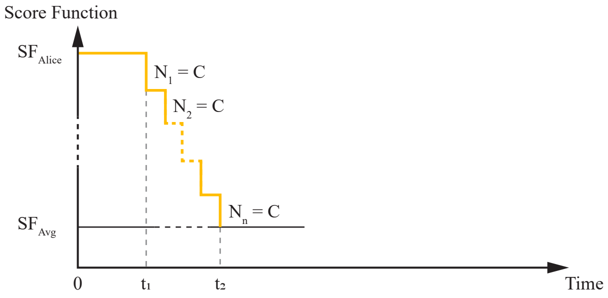

Figure 6. First type of Attack

1. User Alice, owning a large number of tokens, waits until her score function achieves its maximum value and creates an $N = SF_{Alice}$ dust transactions from a single wallet, where $SF_{Alice} \gg C$. Each user with the maximum score function value lower or equal to SF will be blocked while current $SF_{Alice} > SF$. It will happen not earlier than after $n = (SF_{Alice} - SF)/C$ blocks (see Figure 6). According to (1), Alice needs about $\alpha \cdot 1/R \cdot S$ tokens to fill next αL block with her transactions, where $\alpha \in [0, 1]$.
2. User Alice creates lots of small wallets and with $SF_{A(i)}$ higher than SF of average users. And generates a series of outputs to all the addresses of sybil nodes with one or more transactions per address. When the transaction is committed from these wallets, Alice captures the bandwidth, forcing the other users wait longer before their transaction is committed to the blockchain.

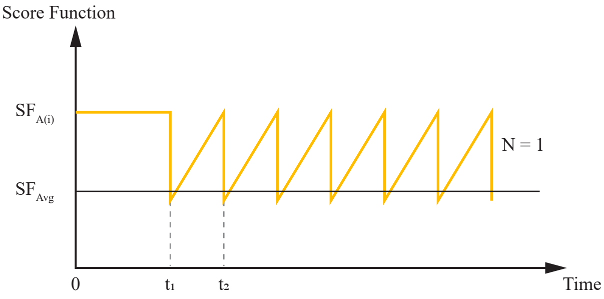

Figure 7. Second Type of Attack

Alice with I wallets $i_1, i_2, \ldots, i_I$ makes transactions among them in such a way that the total number of incoming tokens are equal to outgoing for each particular wallet. As a consequence, even though the SF of the first wallet reduces less than that of the next user's SF, in a short span of time Δt, where $\Delta t = t_2 - t_1$, the SF is increased back to initial SF. In other words, Alice needs about $\alpha \cdot 1/R \cdot S$ tokens to capture $\alpha \in [0, 1]$ relative throughput.

In both boundary cases Alice freezes the same relative throughput on average with the same total amount of tokens on her wallets. It is directly proportional to the amount of bandwidth captured α and inversely proportional to the reserve ratio R.

V. CONCLUSION

In this paper, the block space fractional reservation for blockchains in terms of a score function is introduced. This concept was successfully implemented as a demo on Exonum framework by creating a private blockchain. The score function only influences mempool processing and could be implemented on different frameworks. The approach could be useful to prevent DoS attacks in public blockchains without transaction fees and for private blockchains without any tokens.

VI. ACKNOWLEDGEMENT

The research of Stanislav Kruglik was supported by RFBR according to the research projects No. 18-07-01427, 18-37-00459, 19-01-00364, 19-37-80006.

References

[1] J. Dollimore, T. Kindberg and G. Coulouris, "Distributed Systems: Concepts and Design", p. 944, 2005.

[2] NCCIC, "Understanding denial-of-service attacks", 2009. [Online]. Available: https://www.us-cert.gov/ncas/tips/ST04-015.

[3] E. Kaspersky, "A Brief History of DDoS Attacks", 2016. [Online]. Available: https://eugene.kaspersky.com/2016/12/06/a-brief-history-of-ddos-attacks/.

[4] A. Lloyd, "The Effects of DDoS Attacks on Essential Services", 2018. [Online]. Available: https://www.corero.com/blog/887-the-effects-of-ddos-attacks-on-essential-services.html.

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", www.bitcoin.org, pp. 1–9, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[6] M. Swan, "Summary for Policymakers", in Climate Change 2013 — The Physical Science Basis, Intergovernmental Panel on Climate Change, Ed. Cambridge: Cambridge University Press, 2015, pp. 1–30.

[7] M. Pilkington, "Blockchain Technology: Principles and Applications", in Research Handbook on Digital Transformations. Springer, 2016, pp. 225–253.

[8] H. M. Kim and M. Laskowski, "Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance", SSRN Electronic Journal, vol. 25, No. 1, pp. 18–27, 8 2016. [Online]. Available: http://www.ssrn.com/abstract=2828369.

[9] T.-T. Kuo, H.-E. Kim and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications", Journal of the American Medical Informatics Association, vol. 24, No. 6, pp. 1211–1220, 11 2017.

[10] S. Angraal, H. M. Krumholz and W. L. Schulz, "Blockchain Technology", Circulation: Cardiovascular Quality and Outcomes, vol. 10, No. 9, pp. 5665–5690, 9 2017.

[11] P. Mamoshina, L. Ojomoko, Y. Yanovich, A. Ostrovski, A. Botezatu, P. Prikhodko, E. Izumchenko, A. Aliper, K. Romantsov, A. Zhebrak, I. O. Ogu and A. Zhavoronkov, "Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare", Oncotarget, vol. 9, No. 5, pp. 5665–5690, 1 2018. [Online]. Available: http://www.oncotarget.com/fulltext/22345.

[12] Y. Yanovich, I. Shiyanov, T. Myaldzin, I. Prokhorov, D. Korepanova and S. Vorobyov, "Blockchain-Based Supply Chain for Postage Stamps", Informatics, vol. 5, No. 4, p. 42, 11 2018.

[13] N. Alzahrani and N. Bulusu, "Block-Supply Chain: A New Anti-Counterfeiting Supply Chain Using NFC and Blockchain", in Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems — CryBlock'18. New York, New York, USA: ACM Press, 2018, pp. 30–35.

[14] Bitfury Group and J. Garzik, "Public versus Private Blockchains. Part 1: Permissioned Blockchains", bitfury.com, pp. 1–23, 2015. [Online]. Available: http://bitfury.com/content/5-white-papers-research/public-vs-private-pt1-1.pdf.

[15] "Public versus Private Blockchains Part 2: Permissionless Blockchains", bitfury.com, pp. 1–20, 2015. [Online]. Available: http://bitfury.com/content/5-white-papers-research/public-vs-private-pt2-1.pdf.

[16] C. Cachin, "Architecture of the Hyperledger Blockchain Fabric", IBM Research, vol. July, 2016.

[17] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus and Future Trends", in 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, 6 2017, pp. 557–564. [Online]. Available: http://ieeexplore.ieee.org/document/8029379/.

[18] D. M. Khaled Baqer, Danny Yuxing Huang and N. Weaver, "Stressing Out: Bitcoin Stress Testing", in Financial Cryptography Workshops 2016, 2016.

[19] M. T. Marie Vasek and T. Moore, "Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem", in Financial Cryptography and Data Security, 10 2014, pp. 57–71.

[20] D. Gilbert, “Blockchain Complaints Hit Record Level As Bitcoin Transaction Times Grow And Fees Rise”, 2016. [Online]. Available: https://www.ibtimes.com/blockchain-complaints-hit-record-level-bitcoin-transaction-times-grow-fees-rise-2332196.

[21] S. Choi and K. G. Shin, “Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks”, in Proceedings of the ACM SIGCOMM ’98 conference on Applications, technologies, architectures, and protocols for computer communication — SIGCOMM ’98, vol. 28, No. 4. New York, New York, USA: ACM Press, 1998, pp. 155–166.

[22] A. B. Abel, B. Bernanke and D. D. Croushore, Macroeconomics. Pearson, 2014.

[23] Steemit, “Steem: An incentivized, blockchain-based, public content platform”. Steem. io, pp. 1–32, 2017. [Online]. Available: https://steem.io/SteemWhitePaper.pdf.

[24] F. Vogelsteller and V. Buterin, “ERC-20 Token Standard”, 2015. [Online]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md.

[25] E. Lombrozo, J. Lau and P. Wuille, “Segregated Witness (Consensus layer)”, 2015. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki.

[26] D. Starodubov, “Steam: Motivated Social Media Blockchain Platform (russian)”, 2017.

[27] Y. Yanovich, I. Ivashchenko, A. Ostrovsky, A. Shevchenko and A. Sidorov, “Exonum: Byzantine fault tolerant protocol for blockchains”, 2018.