Transaction Finality through Ledger Checkpoints

Ratul Antik Das
Research and Development
Kona Software Lab
Dhaka, Bangladesh
ratul.antik@konasl.com

Md. Muhaimin Shah Pahalovi
Research and Development
Kona Software Lab
Dhaka, Bangladesh
muhaimin.shah@konasl.com

Muhammad Yanhaona
Research and Development
Kona Software Lab
Dhaka, Bangladesh
nur.yanhaona@konasl.com

***Abstract*—**

***Index Terms*—Computer Networks, peer-to-peer computing, distributed information systems**

I. Introduction

Since the publication of Satoshi Nakamoto's 2008 seminal paper [26] that begot it, the blockchain technology has spurred a flurry of commercial activities. Initially the interest on blockchain technology was solely due to its prospect as a harbinger of alternative currency systems maintained by the masses. Blockchain based digital currencies, called *cryptocurrencies*, gained significant momentum in the wake of the 2008 global financial crisis [5]. Following the trail of Bitcoin [26], the first cryptocurrency, several cryptocurrencies [2] [17] [30] [4] [8] with their own blockchain networks appeared in the market.

Subsequently, Ethereum [31] demonstrated that a blockchain network can be utilized as a decentralized state machine that serves not only as a maintainer of a distributed ledger of cryptocurrency transactions but also a trusted computation engine for generic computer codes called *smart contracts*. The logic encoded in a smart contract can define the rules of interactions among mutually untrusted parties and subsequently enforces them during the contract execution, thus eliminating the chance of a dispute. The idea of enforcing business negotiations using a computer code was decades old [29] but did not get traction due to a central missing piece: there was no global computer whose execution result could be trusted. The discovery of the blockchain smart contract solved that problem and led to widespread interest in the technology from numerous application domains. Some pundits even hail the blockchain technology as a revolution that can transform the global economy [7] [9].

From a technological standpoint, a blockchain system is a decentralized network of peers who maintain the state of a distributed ledger – *the blockchain* – by each executing/validating all the transactions sent to it. The current state of the ledger is updated by adding a block of transactions. Each block has a pointer to its previous block; hence forming a chain called the blockchain. In Nakamoto's proposal [26], a peer who wants to append a valid block in the blockchain has to solve a computational puzzle whose solution is difficult but verification is cheap. The solution of the puzzle becomes a part of the block the peer is trying to append in the blockchain. This serves as a *Proof of Work* (PoW) [10] of the peer to the rest of the network. Combined with the blockchain data structure, PoW provides a simple basis for concensus [11] among the network peers about the state of the ledger.

Users of a peer-to-peer network send transaction requests to arbitrary peers (or subset of peers). It is theoratically impossible to ensure that all transactions will reach all peers within a defined time interval given that even network connectivity among the peers cannot be guaranteed [19]. Consequently, a deterministic ordering of the requested transactions is also unacheivable. Given this problem, each peer of the blockchain network arbitrarily picks among the transactions it received from the users for processing and tries to add blocks in its local version of the chain using those transactions. Whenever it receives a valid blockchain from any of its network neighbors that is longer – i.e., has more work being done on it – than its own version, it accepts the longer chain as canonical and tries to redo its transactions in that longer chain.

Recent alternatives to PoW based consensus for public blockchain networks such as *Proof of Stake* (PoS) [3] or *Proof of Elapsed Time* (PoET) [15] work under the same principal. These protocols change the block construction process and the critera for deciding the best among candidate blockchains. The chance that a better blockchain will replace the current local chain of a peer remains regardless of their protocol differences. Consequently, the long term persistance of blockchain transactions is always a probabilistic guarantee. Currently, guaranteed permanence – commonly called *transaction finality* – in blockchain technology is available only under stringent trust and connectivity contraints among the network peers [16], [24]. These constraints are typically difficult to apply in scalable, peer-to-peer, public networks.

However, the probabilistic permanence of transactions is a serious hindrance for blockchain technology adoption in modeling real-world business and financial interactions. This is because, unlike blockchain transactions, real-world consequences are typically irreversible or non-replayable. If a product is sold, it is sold. If a service is given, it is given. If an aggrement is signed, it is signed. Reversals of consequent payment transactions in the blockchain cannot nullify them. Hence, the transaction finality problem must be solved to realize blockchain technology's potential in documenting real-

This research is funded by KONA Software Lab, Limited.

world interactions and for cryptocurrencies' acceptance in all kinds of payments.

This paper describes the transaction finality solution of the *Kona blockchain platform*: a blockchain solution supporting smart contracts with additional features needed for porting financial and business applications on a peer-to-peer blockchain network incentivised by PoW mining. Finality is achieved by periodically establishing network-wide consensus about *accepted, irreversible state* of the blockchain ledger we call a *checkpoint*. All transactions up to the latest checkpoint are final, thus, can be safely used for any real-world decision making. Transcations mined into blocks after the latest checkpoint are only probabilistically secure until they are included into the next checkpoint.

The checkpoint establishment process, we call *the checkpoint protocol*, neither compromises the PoW mining protocol nor does it necessitate any new network connectivity constraints among the mining peers. Miners' collaboration during checkpoint establishment is incentivised by a PoW mining based voting scheme. The only addition to the network is a *byzantine fault-tolerant* (BFT) [23] support service that disburses checkpoint ballots, seals checkpoint blocks, and does nothing else. We also prove that under the current PoW incentive scheme, the addition of such a service is mandory for ensuring transaction finality in a peer-to-peer network. It can be proven that our checkpoint protocol is fair, non-exclusionary, and can only be interrupted by a successful DDoS attack [32] on the support service.

The paper provides a discussion of the design motives and a detailed analysis of the qualititative properties along with the description of the checkpoint protocol. In addition, the paper illustrates how a byzantine fault-tolerant support service can be realized and properly incentivised so that it cannot willfully bias or destabilize the checkpoint protocol. The rest the paper is organized as follows:

A. Paper Organization

Section II describes related work on transaction finality in blockchain technology; Section III analyses a PoW incentive scheme's impact on miner collaboration and consequent challenges in achieving transaction finality; Section IV then establishes the model for the check-pointing problem and elaborates on modeling objectives; Section V presents our checkpoint protocol; Section VI analyses how the protocol meets the objectives of Section IV; Section VII then touches on some implementation concerns, in particular, it describes how to implement a BFT support service; Finally, Section VIII concludes the paper along with a discussion on future improvements to the protocol.

II. Related Work

It has been theoretically proven that purely peer-to-peer blockchains are probabilistic state machines that cannot provide any finality guarantee [28]. The proof follows easily from the fact that malicious peers can reach 51% majority in a network and reverse all transactions. So any discussion of blockchain ledger stability assumes that the majority mining power is held by the honest peers. A more practical analysis [27] shows that assuming a bounded network delay and no message loss, a blockchain network achieves consistency (i.e., transaction finality) with probablity approaching 1. The problem is messages can be lost and the network delay bound cannot be predermined in practical peer-to-peer network even with majority honest peers. Hence, *probabilistic* transaction finality is the best that can be achieved without any new restriction on the blockchain network.

However, existing blockchain consensus protocols that claim to provide transaction finality impose restrictions on network participation that are difficult to realize in a peer-to-peer setting where participants can join and leave at any time. For example, the Ripple [16] protocol assumes that 80% neighbors of each honest peer are honest and malicious neigbors cannot intercept communications between honest peers. Similarly, the Stellar consensus protocol [24] assumes that the majority honest peers of the network form a connected sub-network that provides guaranteed message delivery. The Tendermint consensus protocol [12] also assumes $\frac{2}{3}$ majority of honest peers receives every message within a bounded time delay. Under this assumption, it provides a *PBFT* [14] solution decentralized for blockchain technology domain. The central issue with all these protocols is that they assume a gossip communication protocol [6] can be utilized as a mean to implement a reliable broadcast primitive [21] in a peer-to-peer network. This assumption is not theoretically valid. Hence these protocols suffer stagnation when broadcast fails.

The forerunner of the *business blockchain movement*, Hyperledger, provides transaction finality for two of its blockchain solutions: *Iroha* [25] and *Fabric* [20]. Both assume there is a BFT ordering service that orders and reliabily delivers messages to all network peers. The network is definitely not peer-to-peer in either case and so far no convincing BFT solution for a distributed ordering service has been proposed.

None of the aforementioned protocols work on a network incentivised by PoW mining. The only notable transaction finality solution for PoW blockchain networks is Ethereum's Casper [13]. Casper assigns validator nodes (based on deposit of stakes) among the mining population for regularly voting on alternative blockchain versions. The version receiving $\frac{2}{3}$ majority of votes is called a finalized checkpoint, thus irreversible. Since the underlying networking characteristics remain the same, $\frac{2}{3}$ validator nodes may fail to communicate, consequently to collaborate, for establishing a voting consensus. Thus, in the worst case no checkpoint can be finalized.

A major difference between ours and existing transaction finality solutions is that we do not assume the existence of any reliable broadcast primitive in the underlying peer-to-peer network. Still, we periodically establish checkpoints in the blockchain ledger. In addition, although we employ voting for reaching consensus about a checkpoint like others, our voting process is also governed by a PoW incentive unlike any.

III. An Analysis of PoW Incentive

From a technical perspective, Nakamoto's Bitcoin [26] took the world by storm because it shows for the first time that a highly secure distributed system can be built from laissez-faire collaboration of a network of mutually-distrusted autonomous peers. Even continuous network connectivity among the peers are not required and the network-wide connection topology remains dynamic. He deemed a PoW for block mining is necessary to thwart traditional networking attacks such as the Sybil Attack [18] on such a system. To successfully alter a shared blockchain ledger of honest miners, an attacker has to surpass their combined mining power to construct an alternative chain that is longer; then offer it to them for synchronization. The attempt become more futile as the population of honest peers increases.

The strength of information security in a blockchain network is proportional to the mining capacity of its honest peers. Therefore, from its inception, keeping honest peers interested in network participation is a central concern in blockchain technology. Nakamoto's ingeneous idea was to make participation in block mining process an economic activity [22] by rewarding a block miner for his/her PoW in terms of transaction fees and a block reward. This is the crux of the PoW incentive scheme.

PoW incentive makes honest behavior the rational behavior [26] in the blockchain network. This particular feat is the source of its resilience. So far no alternative to PoW incentive is proven to be equally scalable, secure, and censorship-resistant. Consequently, the largest and most-powerful blockchain networks are still PoW networks despite widespread scrutiny of PoW mining power consumption cost [1]. Hence, transaction finality solutions should also target PoW mining based blockchain networks.

Interestingly, in a PoW mining based blockchain network a consensus about an irreversible ledger state (i.e., transaction finality) cannot be established without halting the mining process, as explained in the following lemma:

Lemma III.1. *Transaction finality is unachievable without halting block mining in a PoW blockchain network.*

Proof. In PoW incentive scheme, all miner activities including computation and communication are governed by rational behavior. Since there is no benefit for a peer in throwing away its local blockchain ledger version and accepting a neighbor's blockchain that has the same amount of PoW being done on it, information about alternative equaly good blockchain versions will not spreed in the network. Consequently, at any instance of time there might be as many equally good blockchain ledger versions as the number of miners. In addition, a peer cannot accurately estimate the number of currently active peers in the network. As a result, it cannot determine if its local version of the blockchain ledger is accepted by the majority. Since a rational miner can convincingly neither deduce if its own attempt to established a network-wide consensus about irrefutable blockchain ledger state will be successful nor deduce whether it will be worthwhile to throw away its own equaly good local ledger state in response to a request from a neighbor, the only rational behavior is to keep lengthening its own local ledger version. In the worst case, all miners will emulate the same behavior and the only consented state will be the genesis state and no transaction can be declared final. Hence halting block mining is essential for forcing a ledger state consensus. □

At what length of the blockchain ledger the next checkpoint consensus should be attempted must be known to all network peers beforehand since otherwise they will not know when to halt in the absence of a reliable broadcast mechanism. A simple way to achieve this is to make checkpoints periodcal to the length of the blockchain or the total amount of PoW. Another alternative is to decide the next checkpoint time as part of the consensus establishement for the current checkpoint.

An important property of transaction finality in a PoW mining based blockchain network is that given it is a requirement, the rational behavior is to halt and establish a checkpoint at intended time before attempting further progress in lengthening the blockchain. This is explained in the following lemma:

Lemma III.2. *The only rational behavior is to collaborate on a checkpoint consensus when a miner's local blockchain ledger reaches the checkpoint length.*

Proof. Without loss of generality, assume that checkpoints are set at intervals periodic to the length of the blockchain ledger. Now contrary to the proposition, assume a miner whose local blockchain ledger has reached the checkpoint length did not to attempt a checkpoint consensus and keeps mining new blocks. Since the rule is to set checkpoints at specific lengths of the blockchain, all blockchain ledger versions that are at least as long as the next checkpoint length are PoW-wise equal for the lagging behind peers regardless of their actual length. Except for the fact that it is rational for a lagging behind peer to accept the ledger version that keeps most of its own mined blocks intact. Since it can happen that the shortest chain that crossed the next checkpoint length is the chain that maximises the profits of the majority miners, the shortest chain can win the next checkpoint. Since all blocks in the loosing chains after the checkpoint length will be dropped, it is not rational for a miner to keep mining blocks after its ledger reaches the checkpoint length without knowing that its own chain maximises the profit of majority lagging behind miners. Which it cannot do without attempting a checkpoint consensus. Thus a contradiction. □

IV. Problem Modeling

V. Checkpoint Algorithm

VI. Fitness Analysis

VII. Implementation Concerns

VIII. Conclusion

A. Future Work

References

[1] Bitcoin energy consumption index. https://digiconomist.net/bitcoin-energy-consumption. Accessed: 2019-02-14.

[2] Bitcoincash: Peer-to-peer electronic cash. https://www.bitcoincash.org/. Accessed: 2019-01-25.

[3] Casper proof of stake compendium. https://github.com/ethereum/wiki/wiki/Casper-Proof-of-Stake-compendium. Accessed: 2019-02-14.

[4] Dogecoin. https://dogecoin.com/. Accessed: 2019-01-25.

[5] Financial crisis of 2007–2008. https://en.wikipedia.org/wiki/Financial_crisis_of_2007%E2%80%932008. Accessed: 2019-02-11.

[6] Gossip protocol. https://en.wikipedia.org/wiki/Gossip_protocol. Accessed: 2019-03-02.

[7] How blockchains could change the world. https://www.mckinsey.com/industries/high-tech/our-insights/how-blockchains-could-change. Accessed: 2019-01-25.

[8] Neo - an open network for smart economy. https://neo.org/. Accessed: 2019-01-25.

[9] The next radical internet transformation: How blockchain technology is transforming business, governments, computing, and security models with mark mueller-eberstein. https://learning.acm.org/webinars/blockchain. Accessed: 2019-01-25.

[10] Adam Back. Hashcash - a denial of service counter-measure. Technical report, 2002.

[11] Michael Barborak, Anton Dahbura, and Miroslaw Malek. The consensus problem in fault-tolerant computing. *ACM Comput. Surv.*, 25(2):171–220, June 1993.

[12] Ethan Buchman, Jae Young Kwon, and Zarko Milosevic. The latest gossip on bft consensus. *CoRR*, abs/1807.04938, 2018.

[13] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. 10 2017.

[14] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 173–186, Berkeley, CA, USA, 1999. USENIX Association.

[15] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). pages 282–297, 10 2017.

[16] D. Stalin David. The ripple protocol consensus algorithm. 2014.

[17] Quinton David. *Dash Cryptocurrency: Why Dash Digital Currency is the Cryptocurrency of the Future and How You Can Profit from It*. CreateSpace Independent Publishing Platform, USA, 2018.

[18] John R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[19] Michael J. Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *FCT*, 1983.

[20] Guerney D. H. Hunt and Lawrence Koved. Blockchain checkpoints and certified checkpoints, May 2018.

[21] Pankaj Jalote. *Fault Tolerance in Distributed Systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[22] Joshua A. Kroll, Ian Davey, and Edward W. Felten. The economics of bitcoin mining , or bitcoin in the presence of adversaries. 2013.

[23] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[24] David Mazières. The stellar consensus protocol: A federated model for internet-level consensus, 2015.

[25] Fedor Muratov, Andrei Lebedev, Nikolai Iushkevich, Bulat Nasrulin, and Makoto Takemiya. Yac: Bft consensus algorithm for blockchain. 09 2018.

[26] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[27] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673, Cham, 2017. Springer International Publishing.

[28] K. Saito and H. Yamada. What's so different about blockchain? — blockchain is a probabilistic state machine. In *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 168–175, June 2016.

[29] Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.

[30] Ikuya Takashima. *Litecoin: The Ultimate Guide to the World of Litecoin, Litecoin Crypocurrency, Litecoin Investing, Litecoin Mining, Litecoin Guide, Cryptocurrency*. CreateSpace Independent Publishing Platform, USA, 2018.

[31] D. Wood. Ethereum: a secure decentralised generalised transaction ledger. 2014.

[32] Shui Yu. *Distributed Denial of Service Attack and Defense*. Springer Publishing Company, Incorporated, 2013.