CS6290: Reading Summary 1

LING Shengchen
Dept. of Computer Science
ID: 57730900

I. Summary of Paper [1]

A. Problem Statement

The paper targets the problem of performance bottleneck in Nakamoto consensus, particularly in two perspectives that 1) only one participant can win the competition while others are discarded as forks, and 2) the slowness is essential to ensure security of blockchain and irreversibility of transactions [2].

B. Problem Significance

The insufficient throughput and long confirmation delay severely limit the development of blockchain techniques, especially for decentralized applications, causing poor user experience, congested network, and skyrocketing transaction fees.

C. State of the Art

The Nakamoto consensus [6] and GHOST rule [2] focus on honest participants converging on one consensus (main chain), while blocks and transactions on "forks" are discarded and result in a waste of computational power and throughput.

To improve performance, previous research focuses on reducing the participations of the consensus, to some extent sacrificing decentralization. For example, Bitcoin-NG [3] forms the consensus via a rotationally elected "leader" but remains the confirmation time slow. Some other proposals, e.g., Algorand [10], elect a "committee" to run Byzantine fault tolerance (BFT) to determine the transaction order, undesirably result in potential hierarchies among participants.

The data structure of direct acyclic graph (DAG) has been introduced to blockchain. For example, Inclusive blockchains [4] extend Nakamoto consensus to DAG and include off-chain transactions consistently. PHANTOM [5] is reasonable but vulnerable to liveness attacks.

D. Contributions

1) *From the perspective of novelty of problem formulation.* The paper summerizes the problem of performance bottleneck into two parts: large amounts of forks and long confirmation time, and seeks a new solution.

2) *From the perspective of novelty of technical solution.* The main idea is that blockchain transactions rarely conflict with each other, thus they can be serialized in any order. Based on this, the paper presents a fast and scalable DAG-based Nakamoto consensus protocol, Conflux, that defers the transaction total ordering and optimistically process non-conflicting transactions in concurrent blocks.

Conflux consensus protocol remains two kinds of relationships between blocks: one parent edge like Bitcoin to achieve irreversibility of ledger, and multiple reference edges simply indicating the "happen-before" relationships, see Fig.1.

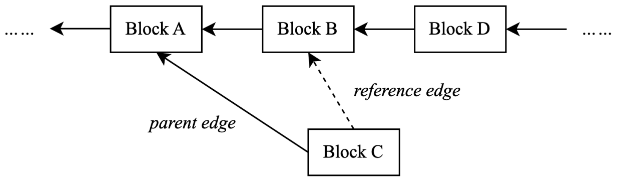

Fig. 1. Two relationships between blocks

Conflux also addresses the problem of ordering blocks in the DAG to meet the total irreversible agreement. The novel ordering algorithm chooses a pivot chain that starts from genesis block and only contains parent edges, those blocks in the pivot chain are partially ordered; and then extends the pivot chain partial order to a total order of all blocks based on the reference edges, which is to say, partitions all blocks into epochs, see Fig.2. The pivot chain is selected by a modified GHOST rule [7].

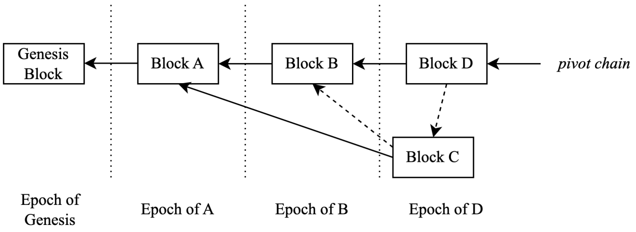

Fig. 2. Pivot chain and epochs

The paper also presents a prototype implementation of Conflux, which is the first blockchain system that uses a DAG-based Nakamoto consensus protocol and can process thousands of transactions per second (TPS).

3) *From the perspective of positiveness of experimental evaluation results.* The experimental throughput of Conflux achieves around 3200 TPS for typical Bitcoin transactions, with 20k full nodes and 20Mbps bandwidth limit for each full node.

The results also show that Conflux can confirm blocks in minutes. The high block generation rate enables faster confirmation time of 7.6-13.8 minutes compared with Bitcoin but is similar to GHOST.

Furthermore, the results also indicate high scalability of Conflux, the consensus protocol is no longer the throughput bottleneck, but the processing capability of individual nodes.

As the bandwidth limit arises (e.g., from 20Mbps to 40Mbps), the throughput arises correspondingly (e.g., from 3200 TPS to 6400 TPS).

E. Remaining Questions

1) The paper only covers the consensus protocol design and implementation, but incentive mechanism and economy model are critically important for a blockchain system's long-term operation. In the whitepaper of Conflux [8], this part has been illustrated.

2) Based on the Proof-of-Work (PoW) consensus mechanism, Conflux is not secure under 51% attack, let alone consider selfish mining situations. In the latest version of Conflux [9], a Proof-of-Stake (PoS) parallel chain has been introduced to confirm finality based on PoW rule, further enhancing security.

3) As for the selection of pivot chain still relies on the GHOST-like algorithm, the confirmation time does not improve significantly.

II. SUMMARY OF PAPER [10]

A. Problem Statement

The paper targets the problem of the conflict between latency and confidence in a transaction in existing blockchains. More precisely, sufficient latency ensures security and irreversibility of blocks and transactions, while on-chain applications that require low latency must compromise on the confidence of confirmation for the sake of performance and user experience.

B. Problem Significance

Low performance (e.g., 7 TPS in Bitcoin) is not user-friendly to on-chain applications. Users have to wait for a confirmation time of up to hours or compromise with security. People are eager to a high-performance blockchain platform as an infrastructure for on-chain interactions.

C. State of the Art

Bitcoin [6] uses PoW to achieve consensus with the sacrifice of performance and confirmation time, and many follow-on cryptocurrencies adopt this approach and inherit corresponding limitations.

Byzantine agreement proposals have been proposed before, for example, in Practical Byzantine Fault Tolerance (PBFT) [11]. PBFT is well-behaved when facing a relatively small set of participants (e.g., dozens), but not when facing large quantities of users (e.g., millions or even billions) [12]. Thus, it's not realistic to apply it to public blockchains directly. Honey Badger demonstrated how to build a cryptocurrency using Byzantine fault tolerance, with a fixed set of servers to achieve consensus [13]. The advantage is the high throughput, but the downside is also obvious: it's no longer decentralized. Bitcoin-NG [3], Hybrid consensus [14], and ByzCoin [15] use Nakamoto consensus to elect leaders, but the risk of "forks" still exists.

Proof-of-Stake (PoS) [16] approach has been proposed to replace PoW to reduce extra waste of computational power. However, in many PoS cryptocurrencies, attackers can still create a fork, it's just holding the risk that once caught, they lose the staked assets.

Other data structures like trees and DAGs are proposed to improve throughput under Nakamoto consensus, which are to be studied whether they will further boost the performance of Algorand.

D. Contributions

1) *From the perspective of novelty of problem formulation.* The paper does not stick to the improvement of the blockchain structure or mining algorithm, but dig deeper to the essence. The paper proposes an idea that the "mining" process aims to make the future blocks unpredictable, in other word, not controlled by any party.

Thus, the paper focuses on finding a cryptography-based method to randomly determine the future blocks, and further extends it to three main challenges (requirements) that the new blockchain, named Algorand, must achieve. Algorand must 1) avoid Sybil attacks, 2) scale to millions of users, and 3) be resilient to denial-of-service attacks.

2) *From the perspective of novelty of technical solution.* The paper proposes the new Byzantine Agreement protocol called BA* and Verifiable Random Function (VRF) to solve the problems mentioned above.

Algorand assigns weights to users based on users' on-chain assets, and BA* periodically chooses a set of users based on weights into a small committee to achieve later consensus, see Fig.3. This ensures most committee members are honest (for Algorand, a fraction of over 2/3 is sufficient).

Further, BA* selects committee members via cryptographic sortition in a private and non-interactive way. Users independently compute a VRF of their private keys and public information from the blockchain to check whether they are chosen or not. If they are chosen, the function returns a string of proof, which can be included in the network messages to prove membership to other users.

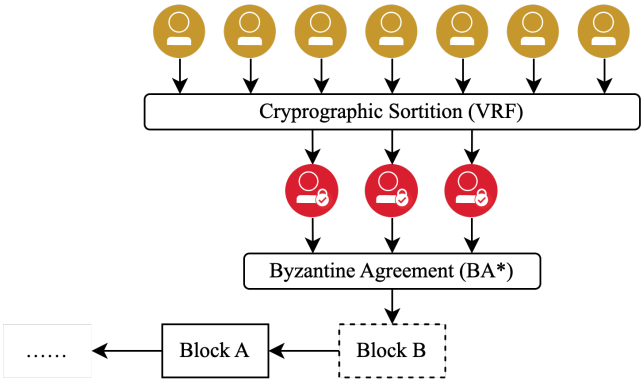

Fig. 3. Algorand consensus process

BA* requires committee members to speak only once to mitigate targeted attacks on them, which is the time when they reveal their membership identity publicly, by avoiding any private state kept by the committee members except for the private keys.

3) *From the perspective of positiveness of experimental evaluation result.* The paper shows that Algorand can confirm a 1 MByte block in ~22 seconds with 5,000 users with great scalability when scaling to 500,0000 users, and has high performance of 125 times the throughput of Bitcoin, and the latency is acceptable even with actively malicious users.

E. Remaining Questions

1) The paper mainly focuses on technical solutions of Algorand, however, the incentive mechanism to encourage users to participate remains to be included. In the latest whitepapers and actual blockchain system [17], it has been proposed and implemented.

2) The BA* is designed to tolerate at most 1/3 dishonest participants, the fault tolerance rate is worse than that in most PoW blockchains, which is normally almost half.

3) After BA* selects a committee member and the message is broadcast to all in the network, the attackers are awared of the identity, as well as other members. In this case, zero-knowledge can help protect privacy identity in a non-interactive way [18].

4) Despite every committee member speaks only once, with the time going, attackers are able to get access to some participants with high weights. When attackers take sufficient control of participants, in some *lucky* rounds, attackers can manipulate the committee on some malicious purposes, e.g., forks.

References

[1] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. C. Yao, "Scaling Nakamoto Consensus to Thousands of Transactions per Second", arXiv:1805.03870v4, 2018.

[2] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin", in *International Conference on Financial Cryptography and Data Security*, 2015.

[3] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol", in P*roceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16)*, 2016.

[4] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive Block Chain Protocols", in I*nternational Conference on Financial Cryptography and Data Security*, 2015.

[5] Y. Sompolinsky, S. Wyborski, and A. Zohar, "PHANTOM GHOSTDAG: A Scalable Generalization of Nakamoto Consensus", https://eprint.iacr.org/2018/104.pdf, 2021.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", https://bitcoin.org/bitcoin.pdf, 2008.

[7] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin", in *International Conference on Financial Cryptography and Data Security*, 2015.

[8] A. Park and A. Veneris, "Conflux Network: Engineering An Economic Design", https://confluxnetwork.org/files/Conflux_Economic_Paper_20201230.pdf, 2020.

[9] F. Long, C. Li, "cip-43: Introduce Finality via Voting Among Staked", https://github.com/Conflux-Chain/CIPs/blob/master/CIPs/cip-43.md, 2021.

[10] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies", in *SOSP '17: Proceedings of the 26th Symposium on Operating Systems Principles*, 2017.

[11] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery", *ACM Transactions on Computer Systems*, 2002.

[12] A. Clement, E. L. Wong, L. Alvisi, M. Dahlin, and M. Marchetti, "Making Byzantine fault tolerant systems tolerate Byzantine faults", in *Proceedings of the 6th Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[13] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The Honey Badger of BFT protocols", in *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS)*, 2016.

[14] R. Pass and E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model", Cryptology ePrint Archive, Report 2016/917, http://eprint. iacr.org/, 2016.

[15] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing Bitcoin security and performance with strong consistency via collective signing", in *Proceedings of the 25th Usenix Security Symposium*, 2016.

[16] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending Bitcoin's proof of work via proof of stake", in *Proceedings of the 2014 Joint Workshop on Pricing and Incentives in Networks and Systems*, 2014.

[17] Algorand White Papers, https://www.algorand.com/technology/white-papers, 2019.

[18] D. Bernhard, O. Pereira, and B. Warinschi, "How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios", *Advances in Cryptology - ASIACRYPT*, 2012.