An Optimized Round-Robin Scheduling of Speakers for Peers-to-Peers-based Byzantine Faulty Tolerance

Guangsheng Yu*, Xu Wang*†, Xuan Zha*†, J. Andrew Zhang*, Ren Ping Liu*
* Network Security Lab, Global Big Data Technologies Centre, University of Technology Sydney
† School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
Email: {guangsheng.yu, xu.wang-3, xuan.zha}@student.uts.edu.au, {andrew.zhang, renping.liu}@uts.edu.au

[1] ***Abstract*—Blockchain technology has been showing its strong performance on decentralized security when integrating with Internet of Things network. However, the trilemma of scalability-security-decentralization exists in Blockchain-based IoT. Therein the typical round-robin scheduling implemented in the Byzantine Faulty Tolerance (BFT) proposed by Neo's Blockchain has a significant delay when consecutive faulty miners exist. This paper proposes a novel analysis model for evaluating the network performance collapse in general, followed by an optimized round-robin scheduling for the case when the mutual latency difference is not significant enough for ranking. Based on the model, the optimized mechanism is able to increase the block rate for a specific subset of consecutive faulty miners by nearly 50% and provide a linearly positive growth rate of the mitigation with respect to the fail rate of a single miner, which strongly promotes the efficiency of the P2P-based BFT consensus algorithm.**

***Index Terms*—Internet of Things, Consensus, Byzantine Faulty Tolerance, Blockchain, Network Performance Collapse**

I. Introduction

Internet of Things (IoT) has been emerged in both academia and industry since early 2000s as one of the core technologies and ecosystems of the Next Generation Network with massive numbers of sensors and actuators widely spread, as well as nodes with powerful computational strength. It is usually regarded as the extension of traditional Internet while an IoT network allows the medium of each node changed from a strong device to a tiny chip or a sensor that can be embedded into wherever necessary. However, there exists a complicated issue that a huge amount of data streams transmitted within an IoT network that consists of billions of IoT devices has significant impacts upon the security of centralized servers. Decentralization is a potential solution where the most difficult challenge is to investigate a method to reach the consensus within a given period in a decentralized manner. That means only a unique result survives.

Blockchain, the kernel of BitCoin [1], is featured with its decentralized tamper-resistance based on a Peers-to-Peers(P2P) network. A P2P network ensures that the entities among the physical network are relatively identical compared with a Client-Server network. Precisely by design, a Blockchain-based database can therefore constitute a trust-free decentralized system. Note that trust-free means the conventional party as an arbitral body is filled in by common cryptographical theorems. This promotes Blockchain to be a suitable role for data recording, storage and identity management, especially for those sensitive data [2]. Bitcoin and Ethereum [3], being famous for the first cryptocurrency that has practically solved double-spending and the first Blockchain that provides decentralized application platform [4], respectively, have been proved their real-world value with the potential in providing tamper-resistant and distributed ledger service. The only possibility to redically attack and destroy a Blockchain system is to have a 51% attack where the number of faulty miners should be more than half. We believe such tamper-resistance property of Blockchain can be of significant value in ensuring trust in IoT systems.

In hopes of converging to a unique result without any forking, a consensus algorithm turns out to be one of the core modules in a Blockchain system. There are some well-known consensus algorithms including Proof-of-Work(PoW), Proof-of-Stake(PoS) and Byzantine Faulty Tolerance(BFT). Practical BFT(PBFT) [5] is thought as the first feasible BFT algorithm to be implemented in live environment. In the context of PBFT, not all the nodes that have joined in the P2P network are able to participate in consensus process unless owning certain ratio of stakes or a valid certificate.

A Blockchain Survey [6] compared Public Blockchain with Consortium Blockchain in terms of multiple properties. Concretely, in terms of **Consensus Determination**, how to define a Blockchain is public or consortium is to distinguish whether a fair and random selection from a set of eligible nodes has been based on their hash power/stake or not, but not "all miners". In terms of **Immutability**, whether a Blockchain system is immutable or not does not depend on its decentralization but the minimum cost spent on destroying the Blockchain system. A Blockchain being consortium does not constitute its less powerful immutability if its market caps is even greater than that of a relatively less popular public chain. In the context of a stable IoT industry, there may be only a small set of nodes owned by multiple business parties to be the miners in a consortium chain, where a round-robin scheduling is usually introduced among these miners to decide a unique miner each round in order to achieve the high scalability. Besides, it is also of importance to prevent the system from data-leaking. Therefore, in the absence of the high level of privacy protection that has not been mature enough to be applied yet, such as

[1]This work was supported, in part, by Ultimo Digital Technologies Pty Ltd under UCOT program.

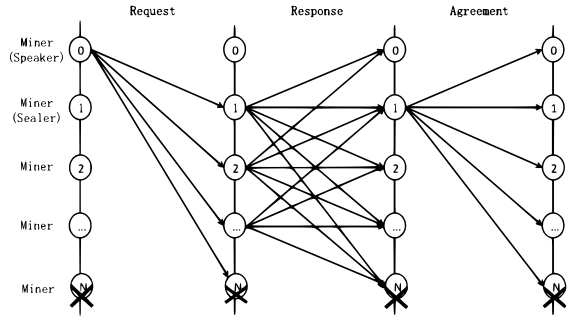


Fig. 1. 3-phase consensus process of P2P-Based BFT.

Ring-Signature [7] and Zero-knowledge Proof [8], a consortium chain is preferable for IoT technologies and PBFT turns out to be an ideal consensus algorithm.

This paper is focusing on an optimized round-robin scheduling used in PBFT. The mechanism solves the latency problem when the latency is uncertain or difference among each sample is not significant enough to be weighed so that ranking will not be easily achieved. It may not behave as expected,

- Even if miners are geographically located closely, the topology of routers in access-layer may still have an unignorable effect.
- The fluctuating is even greater than the difference among each sample.

The rest of the paper is organized as follows. Section II discusses Neo's work on PBFT. In Section III, the problem of network performance collapse in PBFT system is stated. In Section IV an optimized system is presented, followed by the simulation and analysis of the optimized system based on the proposed novel model in Section V. In Section VI, conclusions and future work are drawn.

II. Review the Consensus System

Castro, M. and Liskov, B. proposed PBFT by which dozens of miners are able to securely achieve consensus with $O(n^2)$ given that $\mathbf{f} \leq \frac{n-1}{3}$ where $\mathbf{f}$ denotes the number of faulty miners while n denotes the total number of miners maintained in a list. The list can be static or dynamic. To simplify the question so that we can focus on the scheduling issue, all workarounds such as election of proposing for or against members (dynamic) will be neglected. In order to integrate PBFT with current Blockchain architecture and make it more compatible, Neo [9], the most famous Chinese Blockchain Project, proposed P2P-Based BFT [9] in which the Client-Server architecture is replaced with a P2P-based architecture so that 5-phase consensus process can be at most simplified to 3-phase in the context of consortium chain, which is shown in Fig. 1. Table I shows the necessary symbols.

A static round-robin list is maintained by all miners so that everyone is aware of the miner who will be $G'(h+1)$ (speaker) by comparing p and its own i.

TABLE I
Description of necessary symbols in Neo's system

Symbol	Description
$\Re$	Region or subset that contains at most **f** consecutive faulty miners
C	Canonical chain
B	Block
V	$ChangeView$ message
G	Actual miner of the block
G'	Potential miner of the block
S	Signature of B
SV	Signature of V
t	Block period
μ	Timer starting from the beginning of a new round, $2^{v+1} \times t$
i	Sequence number of each miner in the static list
n	Total number of miners in the list
h	Current height of the canonical chain
v	View number, increased by 1 if a $ChangeView$ phase starts, return to zero if B(h+1) $\in C$
p	$(h-v) \bmod n$
f	Maximum number of faulty miners can be tolerated, $\frac{n-1}{3}$

1) **Request**: The speaker proposes $B(h+1)$ with $S_p(h+1)$ that will be broadcast to all other delegators ($i \neq p$) afterwards. Also $v(h) = 0$.
2) **Response**: Any i has received a $Request$ message and verified the validity, $B(h+1)$ will be broadcast to all other miners including both speaker and other delegators with $S_i(h+1)$.
3) **Agreement**: Any i has received and validated $Response$ messages from no less than $n-\mathbf{f}$ different i, it will immediately broadcast an $Agreement$ attached with $B(h+1)$ and $S_i(h+1)$ collected from $n-\mathbf{f}$ different i, by which "Negotiation Signatures" can be generated for the validity of $B(h+1)$. Also $G(h+1) \leftarrow G'(h+1)$, indicates this round has been successful, B(h+1) $\in C$. A new round will start and miners move back to $Request$ phase.
4) **ChangeView**: If $Agreement$ phase cannot be achieved before μ, i moves to $ChangeView$ phase and broadcasts a $ChangeView$ message to all other miners including both speaker and other delegators with $SV_i(h)$. Any i has received and validated $ChangeView$ messages from no less than $n-\mathbf{f}$ different i, $v(h) \leftarrow v(h)+1$ will be broadcast to all miners. Finally the whole consensus process moves back to $Request$ phase, and decides the new $G'(h+1)$ by comparing new p and i.

Note that around a dozen or two of miners can be thought as the most usual number of miners contained in the static list for most PBFT systems. Therein, $n = 21$ is the number of delegators that participate in a certain round in DPoS-BFT of EOS [10], which is being thought as the most practical value with the lower minimum of efficiency. Both Neo's P2P-based BFT and the optimized mechanism proposed in this paper are subjected to this rule. All analysis and simulation in the rest of this paper also follow this rule.

EOS Blockchain aims for a round-robin scheduling based on a parameter where geography and latency are weighted

so that honest and available $G'(h+1)$ can be regularly decided. What will lead to is the probability that falling into $ChangeView$ phase can be tremendously reduced. Indeed, it is of an appropriate approach if miners are separated globally as multiple subsets [10]. However, as previously stated in Section I, latency becomes too small and random to be the basis to decide $G'(h+1)$ if miners locate closely with each other, for which another feasible solution that modifies the mechanism of $ChangeView$ phase is necessary.

Therefore, it is worth investigating the circumstance that a set of consecutive miners being faulty, which can be happening very likely in living environment. Given some examples as below,

- Multiple processes launched on one single cloud instance will all be killed once the instance is somehow turning faulty.
- Multiple sets containing more than one faulty miners usually exist if $\mathbf{f}$ is big enough.

There exists a network performance collapse in P2P-based BFT system when multiple consecutive miners become faulty. This paper proposes a novel model to analyze the network performance collapse followed by the optimized system that improves the network performance in the context of the proposed model, as well as its evaluation.

III. Problem Statement and Modelling Analysis

Note that the case in which $\mathbf{f} > \frac{n-1}{3}$ is neglected for the rest of this paper, as no new block can be generated and the process will indefinitely get stuck in the $ChangeView$ phase. For all cases in which $\mathbf{f} \leq \frac{n-1}{3}$, a novel analysis model is proposed to evaluate the network performance collapse as follows.

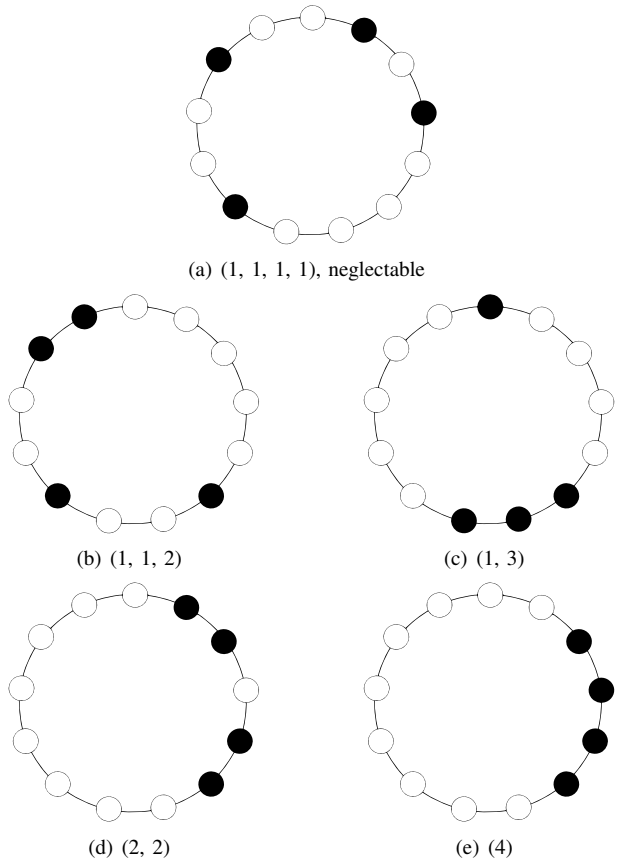
A. A Novel Analysis Model

$\mathbf{f}$ can be separated into multiple combinations of $[f_1, f_2, \ldots, f_m]$ via $\Phi(P, n, f)$, indicating that there are m subsets of faulty miners. Note that $\Re$ can be one of the combinations or exactly $\mathbf{f}$ itself. Also note that P denotes the Partition function P [11], while Φ denotes a vector consisting of the probability that each possible combination happens, excluding the combination with all-ones. f_i consists of one or more consecutive faulty miners. An example where $n = 13$, $\mathbf{f} = 4$ is given as follows,

$$\Phi(P, n, f) = \left[\frac{C_{n,1}}{C_{n,4}}, \frac{C_{n,1} \times C_{n-5,1}}{C_{n,4}}\right] \vee \left[\frac{C_{n,1} \times [C_{n-4,2} - C_{n-5,1}]}{C_{n,4}}, \frac{C_{n,1} \times \frac{C_{n-5,1}}{2}}{C_{n,4}}\right]$$

$$P(f) = P(4) = 5 \tag{1}$$

The vector in (1) contains the probability that each of $[(4), (1,3), (1,1,2), (2,2)]$ happens. The corresponding model with all combinations is shown in Fig. 2. It indicates that there exist 4 combinations excluding all-ones, which matches with $P(4) - 1 = 4$.



(a) (1, 1, 1, 1), neglectable

(b) (1, 1, 2)

(c) (1, 3)

(d) (2, 2)

(e) (4)

Fig. 2. All combinations if $n = 13$, $\mathbf{f} = 4$.

This model is general for a round-robin scheduling in any consensus systems given that,

$$\begin{cases} \mathbf{f} \leq \dfrac{n-1}{3}; & \text{(2a)} \\ G(k) = G(k+n), \\ k \text{ is the first successful block number when starting} \\ \text{to observe the model.} & \text{(2b)} \end{cases}$$

B. Neo's System

Given that the previous model, there arises a question. How long does it need to take to step over $\mathbf{f}$ given that $\mathbf{f} \leq \frac{n-1}{3}$ if the original Neo's algorithm is being used? It can be solved by obtaining the delay $\mathcal{T}$ and block rate $\mathcal{R}$ via the following process. Firstly, the consensus process is shown in Fig. 3.

Note that whoever keeping a token can be the potential speaker of current round. The lower right-arrows with $h = h_0$ starting from m to n indicate that a miner with $i = m$ passes the token to a miner with $i = n$ as n being the potential speaker of the round of $h = h_0$. The upper left-arrows with $v(h) = v'$ starting from n to m indicate that a miner that is the current potential speaker with $i = n$ passes the token back to a miner with $i = m$ as m being the next potential speaker of the round of $h = h_0$ with the view number v increasing to v'.

As shown in Fig. 3, the arrow steps backward, that is,

$$p(h, v(h)) \leftarrow (p(h, v(h)) - 1), \quad \text{if } v(h) \leftarrow v(h) + 1; \tag{3}$$

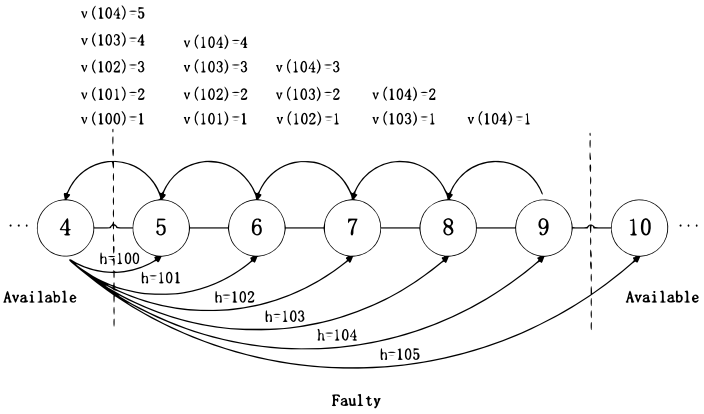


Fig. 3. The consensus process of P2P-based BFT of Neo's Blockchain.

$$p(h+1, v(h+1)) \leftarrow (p(h,0)+1)\,, \quad \text{if } B(h+1) \in C \wedge v(h+1) \leftarrow 0. \tag{4}$$

In the case of $\mathbf{f} \leq \frac{n-1}{3}$, $\mathbf{f}$ can be separated into multiple combinations of $[f_1, f_2, \ldots, f_m]$. There are m subsets of faulty miners and each of them contains f_i consecutive faulty miners. The system needs to take the following delay to step over all faulty miners and start to generate $B(h+f)$,

$$\sum_{v(h)=0}^{f} \mu = t\sum_{l=1}^{f}\sum_{k=1}^{l} 2^k = 2t\sum_{l=1}^{f}(2^l - 1) = 2t(2^{f+1} - 2 - f) \tag{5}$$

$$\mathbb{T}_f^{original} = 2t(2^{f+1} - 2 - f) + (f-1)t, \quad f > 0 \tag{6}$$

$$\mathcal{T} = \sum \mathbb{T}_f, \quad \text{where } \mathbf{f} = \sum_{i=0}^{m} f_i, \quad \mathbb{T}_0 = t \tag{7}$$

$$\mathcal{R} = \frac{\mathcal{T}}{n} \tag{8}$$

$\mathbb{T}_0$ equals to a normal block period. (6) implies that the original Neo's algorithm has a network performance collapse with $O(2^{n+1})$. Therein, $\mathcal{T}$ in (7) denotes the total delay to step over n rounds with block rate $\mathcal{R}$ shown as (8). Note that stepping over n rounds is equivalent to stepping over $\mathbf{f}$ if the delay for the rest of non-faulty miners are treated as zeros in each of the combination set, that is $\mathbb{T}_0 \times (n - |\mathbf{f}|)$. Such a network performance collapse is unacceptable for a consortium chain system.

IV. Optimize the Consensus System

TABLE II
Description of additional symbols in the optimized system

Symbol	Description
k	Counter to skip the leftover consecutive faulty miners in $\Re$. Note that $\Re_{[i,i+\mathbf{f}-1]}$ is equivalent to f_i with $\mathbf{f}$ faulty miners. $\mathbf{f} = f_i = \Re$ is the worst case, that is [(4)] in Fig. 2.
p	$(h + v + k) \bmod n$, where $k(0) = 0$.

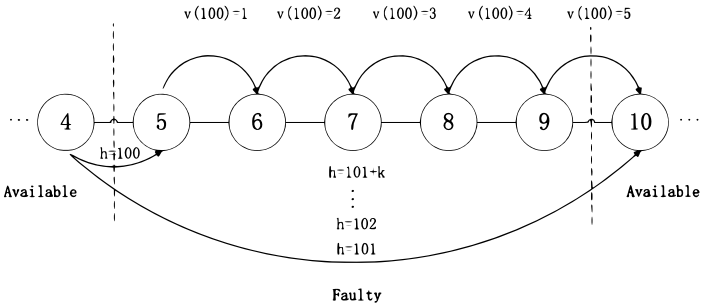


Fig. 4. The optimized consensus process.

The consensus process is shown in Fig. 4, as well as the additional symbols is shown in Table II.

Note that all symbols denoted in Fig. 4 follow the same rules as Fig. 3, except that the upper left-arrows turn to be rightward. As shown in Fig. 4, the arrow steps forward until reaching the first available miner $i' \notin \Re_{[i,i+\mathbf{f}-1]}$. In other words, $i' \notin f_i$ when introducing the model proposed in Section III. The arrow comes to a standstill for $k+1$ rounds until $v(h) = 0$. Even if it is in the worse case that $\Re_{[i,i+\mathbf{f}]} \subset \emptyset$ while $\left|\Re_{[i+\mathbf{f},i+2\mathbf{f}-1]}\right| = |\mathbf{f}|$ and $i' \in \Re_{[i+\mathbf{f},i+2\mathbf{f}-1]}$ (In our model, it can be denoted as $f_i' \leftarrow f_i$), μ will not be indefinitely diverged. It is formulated as follows.

$$k(h+1) = \begin{cases} (\max\{0, k(h)-1\})\,, & \text{if } v(h) = 0; \quad \text{(9a)} \\ (v(h) - 1 + k(h))\,, & \text{otherwise.} \quad \text{(9b)} \end{cases}$$

p and v behave as (3) and (4), with an additional counter k. Given that $k(0) = 0$, when $B(h+1)$ is successfully generated, k stores the state of v if $v \neq 0$ so that μ becomes independent, as shown in (9a). On the other hand, k decreases by one every round given that $k \in \mathbb{N}$. Thus the arrow is prevented from moving backwards to faulty miners and comes to a standstill until $k = 0$, as shown in (9b).

In the case of $\mathbf{f} \leq \frac{n-1}{3}$, $\mathbf{f}$ can be separated into multiple combinations of $[f_1, f_2, \ldots, f_m]$. There are m subsets of faulty miners and each of them contains f_i consecutive faulty miners. The system needs to take the following delay to start to generate $B(h+f)$,

$$\mathbb{T}_f^{optimized} = \sum_{v(h)=0}^{f} \mu = t\sum_{k=1}^{f} 2^k = t(2^{f+1} - 2), \quad f > 0 \tag{10}$$

(10) implies that the optimized mechanism mitigates the network performance collapse to the one with only $O(2^n)$, which can promote a specific subset of consecutive faulty miners with a decrement of nearly 50%. The corresponding delay $\mathcal{T}$ and block rate $\mathcal{R}$ can be obtained by (7) and (8). Note that due to the fact that $\mathbb{T}_1^{original} = \mathbb{T}_1^{optimized}$, $\mathcal{T}$ is independent to the change of the mechanism with all-ones combination. Therefore $f_i = [1,1,1,1]$ in Fig. 2 is neglectable.

In addition, Algorithm 1, Algorithm 2 and Algorithm 3 show the implementation in detail.

- Algorithm 1 shows the complete process of $ChangeView$

phase and how it affects the election of the current speaker. It is shown that an infinite for-loop is being activated for a current mining work. Therein a concurrent channel is being opened for the timer of $ChangeView$, while the election is in progress by comparing p and i outside the channel. Every time the mining work is initialized or a $ChangeView$ is active, Algorithm 2 is invoked from which a new v and p can be obtained. Once $B(h+1)$ is successfully generated and broadcast, Algorithm 3 is invoked.

- Algorithm 2 implements $p = (h + v + k) \bmod n$.
- Algorithm 3 implements (9a) and (9b).

The proposed optimized mechanism can prove the generality of the analysis model in Section III by providing the performance simulation followed by the analysis in Section V. According to the result of the analysis, the improvement of the mechanism can also be proved.

V. Performance Simulation and Analysis

In this section, Golang is used for simulations due to its strong performance on concurrency and distributed architecture. Ethereum platform whose client is also written by Golang is used in our simulation environment. In our simulations, Fig. 5 and Fig. 6 are simulated via the testnet of Ethereum platform. Fig. 7 is simulated via a Golang-based ring structure. The following simulations are conducted and followed by the corresponding interpretations.

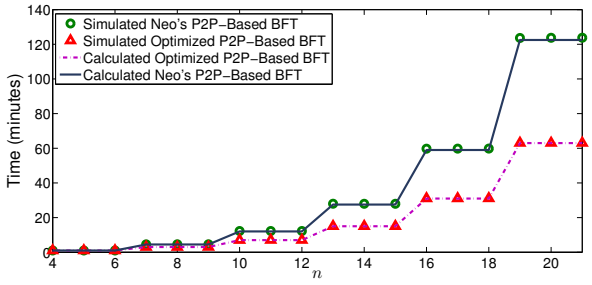


Fig. 5. Comparison between Neo's and Optimized P2P-Based BFT in terms of the maximum **f**.

Fig. 5 shows Neo's and Optimized P2P-Based BFT in terms of the maximum **f**. $\mathbf{f} = \frac{n-1}{3}$ as n gets increased. It is shown that the time consumption on stepping over **f** of the optimized mechanism proposed in this paper falls behind that of Neo's mechanism with an increased rate as n gets increased. Due to the number of n getting restricted to a range around dozens, it can be concluded that the optimized mechanism is able to increase the efficiency for a specific subset of consecutive faulty miners by nearly 50%.

Fig. 6 shows Neo's and Optimized P2P-Based BFT in terms of a fixed $n = 21$ and an increasing **f**. That is $\mathbf{f} \in [1, \frac{n-1}{3}]$. It is shown that Neo's mechanism has an order of $O(2^{n+1})$ while that of the optimized mechanism proposed is $O(2^n)$.

Fig. 7 involves the fail rate of a single miner. It is shown that in Fig. 7(a), both the weighted block rate of Neo's mechanism

Algorithm 1: Process of $ChangeView$

Input: $blockPeriod$ denotes the period a speaker needs to sleep to wait for the synchronization. $view$ denotes the view number that is initially set to be zero. N denotes the number of miners contained in the static list. $index$ denotes the index number of the local miner.

Output: A new block with its height number h, denoted as $B(h+1)$.

1 $Assert(view = 0)$
2 $k \leftarrow Block(h).k$, where $Block(0) = 0$
3 $Timeout(timer, start, view, blockPeriod) \leftarrow timer > start + 2^{view+1} \times blockPeriod$
4 **for** $Mining(h+1) = TRUE$ **do**
5 $Parameters \leftarrow NewRound(view, k)$
6 **while** $\star\ TRUE$ **do**
7 $start \leftarrow Time().Now$
8 $timer \leftarrow Time().Timer$
9 **if** $Timeout(timer, start, view, blockPeriod)$ **then**
10 /* $ChangeView(view)$ opens a new concurrent channel in which
11 " **if** $Timeout(Time().Timer, Time().Now, view+1, blockPeriod)$ **then return** $TIMEOUT$ **else return** $DONE$ " */
12 $view_{new}, status \leftarrow ChangeView(view)$
13 **if** $status = TIMEOUT$ **then**
14 $view_{new}, status \leftarrow ChangeView(view_{new})$
15 **else if** $status = DONE$ **then**
16 $Assert(view_{new} = view + 1)$
17 $view \leftarrow view_{new}$
18 $Parameter \leftarrow NewRound(view, k)$
19 **close** all concurrent channels
20 **for** $Parameters$ has not been renewed **do**
21 **if** $index = Parameters.p$ **then**
22 Local miner is a speaker. Propose a new Block.
23 $UpdateK(Parameters)$
24 **else if** $index \neq Parameters.p$ **then**
25 Local miner is a delegator. Wait for a new Block.
26 $UpdateK(Parameters)$
27 $Block(h+1).k \leftarrow k$ and **final**
28 Note that **while** $\star$ indicates a concurrency loop.

Algorithm 2: Process of $NewRound$

Input: $view$ denotes the view Number at current round. k denotes the new memory counter.

Output: The new set of $Parameters$.

$Parameters.v$, $Parameters.p \leftarrow view$, $(h + view + k) \mod N$

Algorithm 3: Process of $UpdateK$

if **return**$(B(h+1))$ **then**
  if $Parameters.v = 0$ **then**
    $k \leftarrow Max(0, k-1)$
  else
    $k \leftarrow k + Parameters.v - 1$
  break $Mining()$ and **close while** $\star$

and that of the optimized mechanism increase as the fail rate gets increased, with the optimized curve having a shift-down and a tiny decreasing of its gradient. It is concluded that there exists an increasing growth rate of difference between the weighted block rate obtained from Neo's mechanism and the optimized mechanism, as the fail rate gets increased, which is shown in Fig. 7(b).

VI. Conclusions

In this paper, we managed to improve the performance of BFT-like consensus algorithms in the context of a consortium-chain-based IoT network without any explicit latency gaps. This paper showed that the round-robin scheduling proposed by P2P-based BFT of Neo's Blockchain has a delay with an order of nearly $O(2^{n+1})$ in $ChangeView$ phase given that $\mathbf{f} \leq \frac{n-1}{3}$. This paper also proposed a novel analysis model for evaluating the network performance collapse in general, followed by an optimized round-robin scheduling when the mutual latency difference is not significant enough for ranking. Based on the model, the optimized mechanism is able to increase the block rate for a specific subset of consecutive faulty miners by nearly 50% and provide a linearly positive growth rate of the mitigation with respect to the fail rate of a single miner,

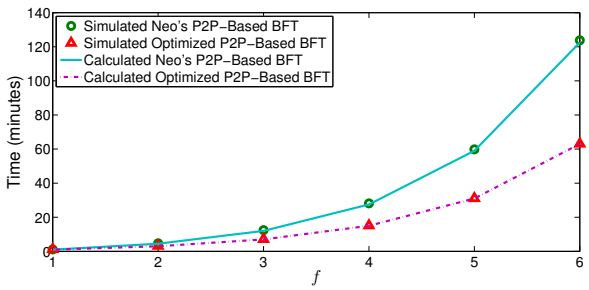
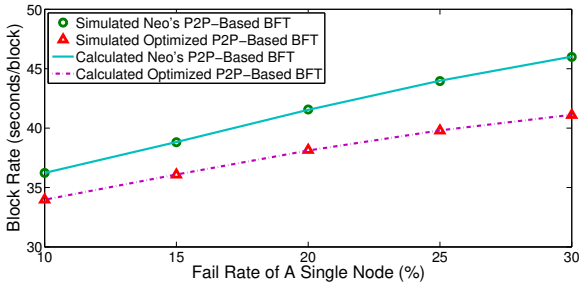
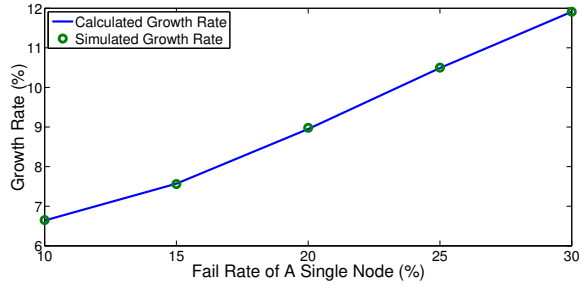


Fig. 6. Comparison between Neo's and Optimized P2P-Based BFT in terms of $n = 21$.



(a) Comparison between Neo's and Optimized P2P-Based BFT in terms of the fail rate of a single miner and the block rate.



(b) Comparison between Neo's and Optimized P2P-Based BFT in terms of the fail rate of a single miner and the growth rate of block rate.

Fig. 7. Comparison with respect to the fail rate of a single miner.

which strongly promotes the efficiency of the P2P-based BFT consensus algorithm. We believe that our proposed analysis model and optimized round-robin scheduling that is based on P2P-based BFT will be helpful for any analysis and potential improvements upon any BFT-like consensus algorithms.

References

[1] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] M. Pilkington, "Blockchain technology: principles and applications. research handbook on digital transformations, edited by f. xavier olleros and majlinda zhegu," 2016.

[3] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[4] K. Bheemaiah. (2015) Block Chain 2.0: The Renaissance of Money. [Online]. Available: https://www.wired.com/insights/2015/01/block-chain-2-0

[5] M. Castro, B. Liskov et al., "Practical byzantine fault tolerance," in OSDI, vol. 99, 1999, pp. 173–186.

[6] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," Work Pap.–2016, 2016.

[7] A. Miller, M. Möser, K. Lee, and A. Narayanan, "An empirical analysis of linkability in the monero blockchain," arXiv preprint arXiv:1704.04299, 2017.

[8] C. Reitwiessner, "zksnarks in a nutshell," 2016.

[9] (2018) NEO White Paper. [Online]. Available: http://docs.neo.org/en-us/

[10] (2018) EOS DPOS-BFT-Pipelined Byzantine Fault Tolerance. [Online]. Available: https://medium.com/eosio/dpos-bft-pipelined-byzantine-fault-tolerance-8a0634a270ba

[11] E. W. Weisstein, "Partition function p," 2002.