Block Access Control in Wireless Blockchain Network: Design, Modeling and Analysis

Yixin Li, Bin Cao*, Liang Liang, Deming Mao, and Lei Zhang

***Abstract*—Wireless blockchain network is proposed to enable a decentralized and safe wireless networks for various blockchain applications. To achieve blockchain consensus in wireless network, one of the important steps is to broadcast new block using wireless channel. Under wireless network protocols, the block transmitting will be affected significantly. In this work, we focus on the consensus process in blockchain-based wireless local area network (B-WLAN) by investigating the impact of the media access control (MAC) protocol, CSMA/CA. With the randomness of the backoff counter in CSMA/CA, it is possible for latter blocks to catch up or outpace the earlier one, which complicates blockchain forking problem. In view of this, we propose mining strategies to pause mining for reducing the forking probability, and a discard strategy to remove the forking blocks that already exist in CSMA/CA backoff procedure. Based on the proposed strategies, we design Block Access Control (BAC) approaches to effectively schedule block mining and transmitting for improving the performance of B-WLAN. Then, Markov chain models are presented to conduct performance analysis in B-WLAN. The results show that BAC approaches can help the network to achieve a high transaction throughput while improving block utilization and saving computational power. Meanwhile, the trade-off between transaction throughput and block utilization is demonstrated, which can act as a guidance for practical deployment of blockchain.**

***Index Terms*—Blockchain, wireless network, CSMA/CA, forking, Markov chain, performance analysis.**

I. Introduction

Wireless blockchain network is proposed to enable a robustness and distributed wireless network for different blockchain applications, such as blockchain-based mobile edge computing [1], [2], and blockchain for vehicles management [3], [4]. Using blockchain to build distributed wireless networks has the following advantages: 1) Alleviate the pressure of high-load nodes in the network and reduce the impact of single point of failure. 2) Improve the security and scalability of network, and reduce maintenance cost, especially for large scale scenarios such as Internet of Things (IoT) [5]. 3) Achieve adaptive matching and behavioral decision-making of users/terminals by involving smart contract [6], [7].

The decentralization and security provided by blockchain for network can be largely attributed to the use of consensus algorithm [5], which motivates the nodes in the network to maintain a single version of the digital ledger without the involvement of a third party. Several consensus algorithms have been proposed, e.g., proof-of-work (PoW) [8], proof-of-stake (PoS) [9], practical Byzantine fault tolerant (PBFT) [10], and Raft [11]. Among them, PoW is the first widely used one in blockchain, and has a better security and node scalability than PBFT and Raft, since the fault tolerance and communication efficiency in PoW is higher than that in PBFT and Raft [12], [13]. In view of these advantages, this work studies the PoW consensus process in wireless network and the analysis can be extended to PoS easily. Using PoW to achieve consensus in wireless network, one of the important steps is to broadcast new block using wireless channel. Under wireless network protocols, the block transmitting will be affected significantly.

Considering the characteristics of wireless network, this work focuses on the consensus process in blockchain-based wireless local area network (B-WLAN) by investigating the impact of carrier sense multiple access with collision avoidance (CSMA/CA), which is a random access mechanism on media access control (MAC) layer. In a typical blockchain design, with ideal communication conditions, the first full node (FN) [14] which generates a valid new block is the winner of bonus. However, as shown in Fig. 1(b), due to the randomness of the backoff counter in CSMA/CA, the first block generated by a FN may not be transmitted immediately, thus the other FNs will keep on mining to generate new blocks. In this case, more than one blocks might be generated during a backoff counter and the latter block has the probability to outpace the first one, thus become the final winner. Meanwhile, the first block will become a fork in the blockchain ledger. The forking problem results in the inconsistency of blockchain ledgers among the FNs, then lead to the waste of computational power and security issues, such as "double-spending" [15]. Due to forking problem, the block generation rate in PoW are slowed down by blockchain system codes and thus the transaction throughput are limited to dozens usually, e.g., 7 transaction

This work was supported in part by National Natural Science Foundation of China under Grant 62071075, and Grant 61941114, in part by the General Project of Natural Science Foundation of Chongqing under Grant cstc2020jcyj-msxmX0704, in part by the Eighteenth Open Foundation of State Key Lab of Integrated Services Networks of Xidian University under Grant ISN20-05, in part by Sichuan International Science and Technology Innovation Cooperation/Hong Kong, Macao and Taiwan Science and Technology Innovation Cooperation Project under Grant 2019YFH0163, and in part by Key Research and Development Project of Sichuan Provincial Department of Science and Technology under Grant 2018JZ0071. *Corresponding author: Bin Cao (caobin@bupt.edu.cn)*

Yixin Li is with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xian 710126, China.

Bin Cao is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xian 710126, China.

Liang Liang is with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China.

Deming Mao is with the China Electronic Technology Cyber Security Co., Ltd, Chengdu 610041, China.

Lei Zhang is with the James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K.

per second (tps) in Bitcoin [8] and 15 tps in Ethereum [16].

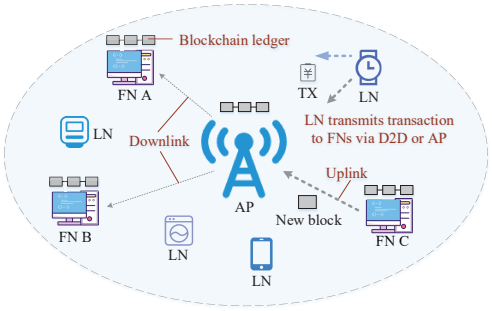
Furthermore, with the evolution of wireless networks, the nodes of network become more dense, and to meet the needs of a surge of service requests, the block generation rate should be accelerated to improve the transaction throughput. In this case, the forking problem will become more serious, since the frequent collisions occurring on the channel prolong the backoff counter and the new block arrives very fast, which increase the forking probability considerably. To address this forking problem and improve B-WLAN performance, we propose mining strategies to reduce the forking probability, and a discard strategy to remove the forking blocks that already exist in CSMA/CA backoff procedure. Based on the proposed strategies, we design four Block Access Control (BAC) approaches to schedule block mining and transmitting effectively. Then, using Markov chain models, we carry out mathematical analysis to show how the different BAC approaches can improve the performance of a B-WLAN. The main contributions of this paper can be summarized as follows.

- We propose mining strategies to pause mining during the backoff and transmission of a new block, which aim to reduce the meaningless computational power consumption on forking blocks and improve block utilization.
- We propose a discard strategy to stop FNs from broadcasting forking blocks, which acts as a key enabler to accelerate block generation rate and improve transaction throughput. This strategy can work in parallel with mining strategies to improve the overall performance of B-WLAN.
- Based on the proposed strategies, we design four BAC approaches and use Markov chain models to conduct performance analysis in B-WLAN. By analysing the stationary probability of Markov chain, we derive the closed-form expressions of key performance metrics, in terms of transaction throughput, block discard rate, block utilization and mining suspension probability.
- Our experimental results validate the effectiveness of mining strategies and discard strategy on improving the transaction throughput and block utilization of B-WLAN. We also make various interesting observations about the impact of blockchain system parameters on the performance trade-off.

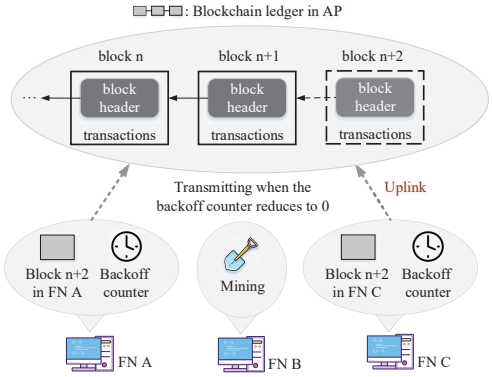
The rest of this paper is organized as follows. Section II describes the consensus process in B-WLAN and forking problem. Section III proposes strategies to address forking problem and improve B-WLAN performance. Based on the strategies, four BAC approaches are designed. Section IV introduces Markov chain models to capture the working process of BAC approaches. Based on the stationary probabilities of the models, the closed-form expressions of key performance metrics are analysed in Section V. Section VI conducts some experiments to compare the performance of four BAC approaches. Then, related works are discussed in Section VII. Finally, we concludes the whole paper in Section VIII.

II. Preliminaries

In this section, we start by introducing the main elements in B-WLAN. Then, we discuss the consensus process and forking problem in the network.



(a) An illustration of blockchain-based wireless local area network.



(b) Channel contention for block transmission.

Fig. 1: An illustration of B-WLAN and channel contention process.

A. Blockchain-Based Wireless Local Area Network

As shown in Fig. 1(a), a typical B-WLAN mainly consists of three elements: lightweight nodes (LNs), full nodes (FNs) and access point (AP). LNs are storage and power-constrained nodes, and the blockchain network allows them to issue transactions without storing the full blockchain ledger [14]. FNs are nodes with enough computing power and storage space, which perform hash operations to generate new blocks for recording transactions. FNs store a full blockchain ledger [14]. For example, in autonomous vehicles application [3], LNs can be the vehicles that perform local tasks, e.g., machine learning. FNs can be vehicle miners that generate new block to collect and share information, e.g., local model updates. In smart factory application [17], LNs can be the factory devices for data collecting. FNs can be the factory computers for data processing and consensus achieving in blockchain.

In blockchain network, the new block of a FN should be broadcast to all the other FNs for achieving consensus. Considering the uncertainty of the geographical distribution of FNs in wireless network, it is difficult to establish device-to-device (D2D) connections for all FNs. In view of this, we consider that AP is responsible for the broadcast of new blocks in B-WLAN, which is more feasible and has a higher block broadcast efficiency than the multi-hop D2D transmission. As shown in Fig. 1(a), when a new block is generated by FN C, it transmits the block to AP through the uplink, and then AP broadcasts the block to all the FNs within its coverage radius through the downlink. During the block transmission process,

AP can store the latest blocks as a backup for download. To enhance the security, one can use more APs to store and broadcast blocks redundantly. However, as a starting point, this work studies the block transmission within a single AP and assumes that the attacker cannot violate communication protocols.

For transactions, the LNs with limited power can transmit transactions to nearby trusted FNs using D2D connections, which requires less energy for transmission. Note that the transactions will be included in a block which broadcast by FNs finally, so the LNs with limited power do not need to broadcast transactions to all FNs from the start. However, if LNs do not trust nearby FNs or the D2D connection is not supported in the environment, LNs can broadcast transactions via AP with a higher cost and longer delay. Since this work focuses on the block transmission, we do not further address the wireless link selection for transactions.

As we know, IEEE 802.11 series is a cost-efficient solution for WLAN that can satisfy most communication requirements in domestic, public, and business scenarios [18]. The primary MAC protocol of IEEE 802.11 is CSMA/CA, which is called distributed coordination function (DCF). In this work, we consider CSMA/CA as the access mechanism for block transmission from FN to AP, and proposes strategies to address forking during block transmission process. Nevertheless, the proposed strategies can work in parallel with the other wireless network protocols in a similar manner. At each block transmission with CSMA/CA, the backoff time is uniformly chosen in the range $[0, W_i - 1]$, where W_i is the contention window and $W_i = 2^i W_{min}$ $(0 \leq i \leq m)$. The value of W_i depends on the backoff stage i (the number of retransmissions). At the first transmission attempt, contention window is equal to the minimum contention window W_{min}. After each unsuccessful transmission, W_i is doubled, up to a maximum value W_{max}. We denote m as the maximum number of retransmissions, where $W_m \leq W_{max}$. A block will be discarded when m-th transmission is unsuccessful.

B. Consensus Process and Blockchain Forking

The main step of the consensus process in B-WLAN can be summarized as follows: 1) The LNs generate transactions and transmit them to FNs through the wireless link. 2) All FNs collect the new transactions and perform hash operations to generate a valid new block. 3) The FN which has a valid new block competes with other FNs based on CSMA/CA for transmitting its block to AP. 4) AP receives and verifies the new block, and then broadcasts it to all FNs. 5) The other FNs receive and verify the new block, then insert it into their local ledgers. If any FN does not receive the new block, it can download block from AP. 6) When AP and most of the FNs have the identical copy of the new block in their local ledgers, the new block and the transactions included in it achieve preliminary consensus successfully. 7) After that, with the accumulating of blocks sequentially, the cost of attack and malicious modification will be increased exponentially [8].

In blockchain consensus process, due to communication delay, more than one blocks at the same height (the position in blockchain) might be created by different FNs, which result in forking problem. To describe the forking problem, we define three working states of a FN: no block, block backoff and block transmitting. Based on the definition, the forking problem will occur when a new block is generated by a FN while the other FNs are in block backoff or block transmitting state.

III. Block Access Control

In this section, we propose mining strategy and discard strategy to address forking problem and improve the performance of B-WLAN. Based on the proposed strategies, we design four BAC approaches to schedule block mining and transmitting.

A. Forking Solution

Mining strategy: The principle of this strategy is to pause mining (hash operations) during the backoff and transmission of a new block, which aims to reduce forking probability and improve the block utilization. Specifically, there are two strategies to pause mining as follows, where strategy I pauses mining based on the block transmission detected on the channel and strategy II pauses mining based on the working state of a FN.

Strategy I: the mining of a FN should be paused whenever a block transmission of the other FN is detected on the channel. The FN resumes the mining when the channel is detected as idle more than a distributed inter-frame space (DIFS)[1]. The reason to pause mining in this case is that when a new block of the other FN is transmitted on the channel, the current mining with the hash of an old block will generate forking blocks or waste computational power. To implement this strategy, an option is to set an additional Flag field in packet header to announce the block transmission. Accordingly, the FN can distinguish the block transmission to pause mining. Meanwhile, the time to pause mining can be easily determined based on the Duration/ID field in packet header, which contains the data transmission time to update the network allocation vector (NAV) in CSMA/CA [19].

Strategy II: the mining of a FN should be paused when the FN generates a new block. The FN resumes mining when the new block is transmitted successfully or discarded. The reason is that a new block might be overtaken by another block with the same height during the random backoff counter, especially when the block generation rate is high. So if a FN in the block backoff state keeps mining, it will generate forking blocks or waste computational power.

Although the mining strategy can reduce the forking probability, it cannot solve the forking problem in B-WLAN thoroughly, since a FN in wireless network cannot detect whether the other FNs are in the block backoff state. In view of this, we propose a discard strategy as follows.

Discard strategy: The principle of this strategy is to discard forking blocks before they are broadcast to the network. Specifically, if a FN receives a same height block transmitted

[1]If a FN generates a block during a DIFS, it can not start the backoff procedure until the end of the DIFS, which increases the forking probability.

TABLE I: Four BAC approaches

	Discard strategy	Mining strategy I	Mining strategy II
BAC-1	✓	-	-
BAC-2	✓	✓	-
BAC-3	✓	-	✓
BAC-4	✓	✓	✓

by the other FN, it should discard its own blocks that do not be broadcast yet. Actually, blocks will be discarded in the following two cases: (i) a FN in the no block state generates new blocks while a successful block transmission occurs on the channel. (ii) a FN in the block backoff state receives a block of the other FN. Using this strategy, the forking blocks will not be broadcast to the network, and thus the hash difficulty of PoW can be very low for accelerating block generation rate and improving transaction throughput.

B. Working Approaches

Based on the proposed strategies, we design four BAC approaches to schedule block mining and transmitting. As shown in Table I, BAC-1 only contains discard strategy; BAC-2 contains discard strategy and mining strategy I, BAC-3 contains discard strategy and mining strategy II, BAC-4 contains all the strategies.

Fig. 2 shows an example of how BAC approaches can work in parallel with CSMA/CA. In this example, we suppose the block is generated simultaneously in four BAC approaches to show when a FN should discard block and pause mining. Accordingly, the block transmission and discard process in Fig. 2(a) are the same in four BAC approaches, since all the approaches contain discard strategy and use CSMA/CA to transmit blocks. We can see that a backoff block of FN B is discarded after the block transmission of FN A is successful. On the other hand, the mining process in Fig. 2(b) is different among four BAC approaches, which is controlled by mining strategy I and strategy II.

BAC-1: only contains discard strategy, thus a FN will keep mining all the time, which is shown in Fig. 2(b). In no block state, if the FN generates a new block while a successful block transmission occurs on the channel, the FN discards its own block based on the discard strategy. If the new block is generated during the other information transmission, collision or channel idle time, the FN schedules its block transmission based on CSMA/CA. Once the channel remains idle more than a DIFS, the FN starts the backoff procedure by selecting a random initial value as the backoff counter. In block backoff state, the FN decreases the backoff counter while listening to the channel. Whenever a block transmission is detected on the channel, the backoff counter is paused and the FN begins to receive a new block. If the block transmission is successful, the FN discards its own block. Otherwise, the FN discards the collision message and continues to listen to the channel. If the channel remains idle more than a DIFS, the backoff counter is resumed. When the backoff counter reaches zero,

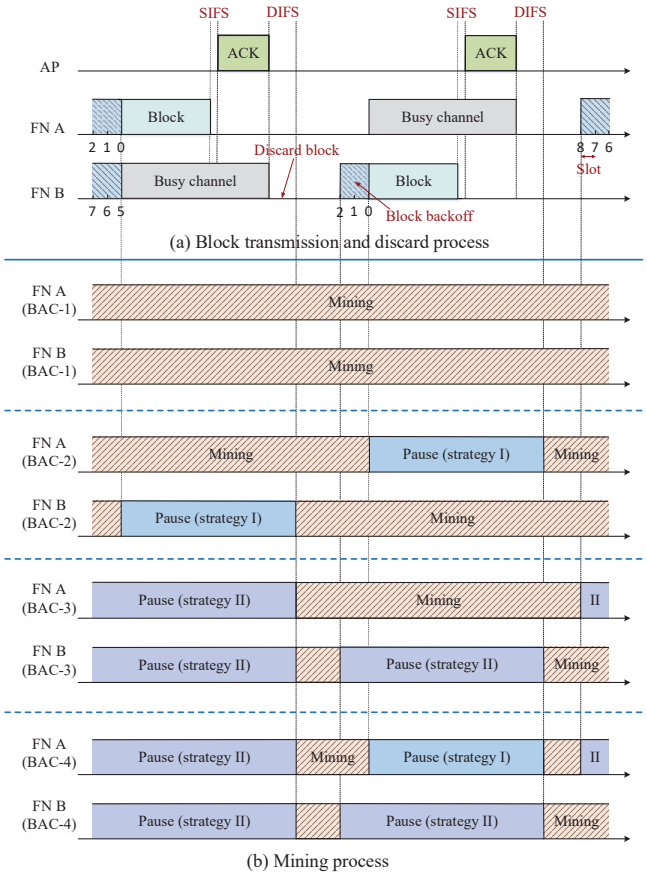


Fig. 2: An example of BAC approaches to work in parallel with CSMA/CA.

the FN enters block transmitting state. If no other blocks and information transmit at this time, the block transmission is successful. Otherwise, the backoff stage increases by one and the FN stays in the backoff state.

Since a FN using the BAC-1 keeps mining all the time, it can generate new blocks during the block backoff and block transmitting states, which results in the queueing of new blocks. In this case, once a FN receives a new block of the other FN, all the blocks in the queue should be discarded. The reason is that the block in queue references the hash of the backoff block, and the discard of a earlier block will invalidate the following blocks in blockchain. Another impact of the mining during backoff and transmitting states is that, if there exist blocks in queue, a FN can directly schedule a new transmission after the earlier transmission is finished.

BAC-2: contains discard strategy and mining strategy I, thus the mining of a FN will be paused whenever a block transmission of the other FN is detected on the channel until the channel is idle more than a DIFS, which is shown in Fig. 2(b). Due to the mining pause, the expected mining time of FN in BAC-2 is less than that in BAC-1, which results in two differences between BAC-2 and BAC-1. The first is that, for a randomly chosen time slot, the probability to leave no block state in BAC-2 is lower than that in BAC-1. The second is that the expected time to generate queueing blocks in BAC-2 is less than that in BAC-1.

BAC-3: contains discard strategy and mining strategy II, thus the mining of a FN will be paused when the FN has a

new block until the new block is transmitted successfully or discarded, which is shown in Fig. 2(b). Since the mining is paused during the block backoff and block transmitting states, block queueing does not exist in BAC-3. So after a new block is transmitted successfully or discarded, a FN will return to no block state and restart mining. Except that BAC-3 has no block queueing, the other behaviors of BAC-3 is similar to BAC-1.

BAC-4: contains all the strategies. There are two differences between BAC-4 and BAC-1. The first is that, for a randomly chosen time slot, the probability to leave no block state in BAC-4 is lower than that in BAC-1. The second is that the block queueing does not exist in BAC-4, thus a FN will return to no block state and restart mining after the FN transmits or discards its block.

IV. Mathematical Modelling

In this section, we formulate the working process of four BAC approaches as Markov chain models to study the stationary probabilities, which act as the basis for performance analysis.

A. Markov Chain Model for BAC-1 and BAC-2

In this work, we analyse the maximum transaction throughput in B-WLAN by assuming that an independent channel is assigned for uplink block transmission. Based on this assumption, the other information is not interfere with block transmission, and we only consider the competition of blocks in the following analysis. Another assumption is that the channel condition is ideal [19], i.e., the only reason of a transmission failure is that a collision occurs on the channel.

We consider a B-WLAN has N FNs. Each FN has three working states: no block, block backoff and block transmitting, which can be depicted as Fig. 3. In this model, the no block state is described by $\{-1, 0\}$; the block backoff states are described by $\{i, k\}$, where $i \in [0, m]$ representing the backoff stage and $k \in [1, W_i - 1]$ representing the backoff counter in time slots; the block transmitting states are described by $\{i, 0\}$ ($i \in [0, m]$), in which the block will be transmitted to channel. When a given FN is in the no block state or backoff state, the channel with probability p_s contains a successful block transmission; the channel with probability p_c contains a collision; the channel with probability $1 - p_s - p_c$ stays idle. Different with this, when a given FN is in the transmitting state, its block will be transmitted to channel, and thus a successful block transmission occurs on the channel with probability $1 - p_s - p_c$; a collision happens with probability $p_s + p_c$. Let T_s be the average channel busy time when a successful transmission occurs on the channel, T_c be the average channel busy time when a collision happens, and σ be the size of time slot. Similar with [19], the Markov chain model adopts a discrete time scale, and one step in this model can be T_s, T_c or σ, which is determined by the channel condition, i.e., the channel may contain a successful transmission, a collision or stay idle.

BAC-1: Using the BAC-1, a FN will perform hash operations with hashrate r (the number of hash operations per second) all the time. Let the hash difficulty of PoW in this network be D representing the expected number of hash operations to find a valid block. Based on [9], we have the time T for a FN to find a valid block is exponentially distributed with block generation rate $\lambda = r/D$. Now we study the behavior of a FN in the no block state and let p_a be the probability that a FN generates a new block and leaves no block state during a step of Markov chain model. In no block state, if the channel contains a successful block transmission, the FN will stay no block state during this step no matter whether the FN generates a new block or not, since all the blocks that do not be broadcast will be discarded by the discard strategy; if the channel contains a collision, the mining time of the FN in this step will be T_c and thus the probability to leave no block state will be $P\{T \le T_c\}$; if the channel stays idle, the mining time will be σ and thus the probability to leave no block state will be $P\{T \le \sigma\}$. In summary, we have

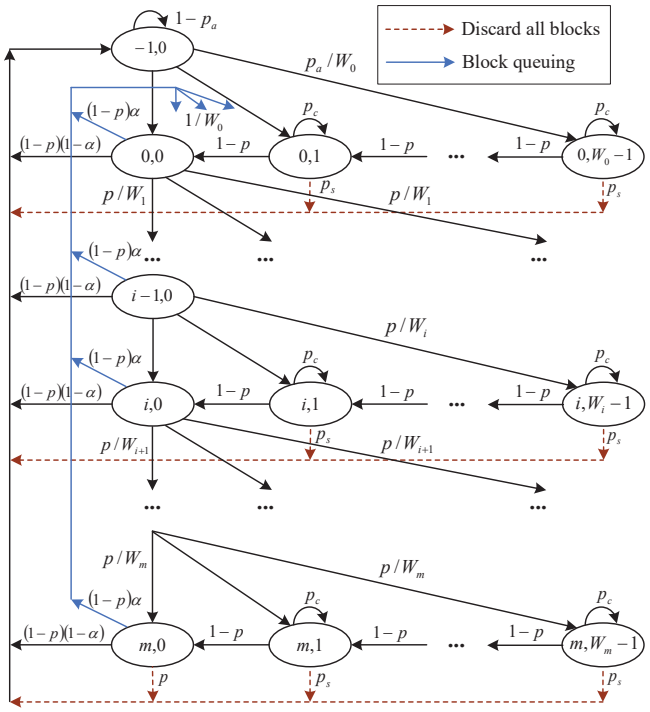


Fig. 3: Markov chain model for BAC-1; When p_a and α change to $\tilde{p}_a$ and $\tilde{\alpha}$ respectively, this model is available for BAC-2.

$$\begin{aligned} p_a =& p_s \cdot 0 + p_c \cdot P\{T \le T_c\} + (1 - p_s - p_c) \cdot P\{T \le \sigma\} \\ =& p_c[1 - \exp(-\lambda T_c)] + (1 - p_s - p_c)[1 - \exp(-\lambda\sigma)]. \end{aligned} \tag{1}$$

Acoordingly, the one-step transition probabilities in no block state can be given by

$$\begin{cases} P\{-1, 0 \mid -1, 0\} = 1 - p_a, \\ P\{0, k \mid -1, 0\} = p_a/W_0, \qquad k \in [0, W_0 - 1]. \end{cases} \tag{2}$$

The FN leaves the no block state with probability p_a and uniformly chooses a backoff time from $[0, W_0 - 1]$. Then, it starts decreasing the backoff counter while listening to the channel. In block backoff states, the FN discards its blocks with probability p_s; the FN pauses the backoff counter with probability p_c; the FN decreases the backoff counter with

probability $1-p$ ($p=p_s+p_c$). Thus, the one-step transition probabilities in block backoff states can be given by

$$\begin{cases} P\{-1,0 \mid i,k\}=p_s, & i\in[0,m],\ k\in[1,W_i-1],\\ P\{i,k \mid i,k\}=p_c, & i\in[0,m],\ k\in[1,W_i-1],\\ P\{i,k-1 \mid i,k\}=1-p, & i\in[0,m],\ k\in[1,W_i-1]. \end{cases} \tag{3}$$

When the backoff counter becomes 0, the FN will enter the block transmitting state $\{i,0\}$ and its block will be transmitted into the channel. If no other FNs transmit block in this slot, the transmission is successful. Otherwise, the backoff stage increases and the FN starts a new backoff counter. Especially, in state $\{m,0\}$, the FN will discard its blocks when a collision occurs. Let p be the probability that a collision is seen by a block transmitted on the channel, where $p=p_s+p_c$.

Now we study the one-step transition probabilities in block transmitting states. Since a FN using the BAC-1 keeps mining all the time, it may generate queueing blocks during the backoff and transmitting states. In this case, when a block discard occurs in block backoff states or state $\{m,0\}$, all the queueing blocks will also be discarded and thus the FN returns to no block state, shown as the dotted lines in Fig. 3. On the other hand, if there exist queueing blocks after a successful transmission in a block transmitting state, a FN will directly move from the block transmitting state to the block backoff state, shown as the blue lines in Fig. 3. Let α be the stationary probability that the queue is not empty after a successful block transmission. Based on [21], $\alpha=\lambda T_q$, where λ is the block generation rate of a FN, and T_q is defined as the expected time of a block spent on backoff and transmitting states counting from a block generation to its successful transmission. Using the expectation formula, T_q can be given by

$$T_q=\sum_{i=0}^{m}p_e(i)\left[(iT_c+T_s)+\sum_{n=0}^{i}\frac{W_n-1}{2}\left(\sigma+\frac{p_c}{1-p}T_c\right)\right], \tag{4}$$

where $p_e(i)$ ($i\in[0,m]$) is the probability that a block exits the backoff scheme through a successful transmission in state $\{i,0\}$, and it can be expressed as

$$p_e(i)=\prod_{n=0}^{i}\frac{1-\left(\frac{1-p}{1-p_c}\right)^{W_n}}{W_n}\left[\frac{p}{1-\frac{1-p}{1-p_c}}\right]^{i+1}\frac{1-p}{p}. \tag{5}$$

The complete proof of T_q in (4) is given in the Appendix. Based on (4), α can be given by

$$\alpha=\lambda\sum_{i=0}^{m}p_e(i)\left[(iT_c+T_s)+\sum_{n=0}^{i}\frac{W_n-1}{2}\left(\sigma+\frac{p_c}{1-p}T_c\right)\right]. \tag{6}$$

Once α is determined, the one-step transition probabilities in transmitting states are

$$\begin{cases} P\{0,k \mid i,0\}=(1-p)\alpha/W_0, & i\in[0,m],k\in[0,W_0-1],\\ P\{i+1,k \mid i,0\}=p/W_{i+1}, & i\in[0,m-1],k\in[0,W_0-1],\\ P\{-1,0 \mid i,0\}=(1-p)(1-\alpha), & i\in[0,m-1],\\ P\{-1,0 \mid m,0\}=(1-p)(1-\alpha)+p. \end{cases} \tag{7}$$

BAC-2: The mining pause during the block transmission results in two differences between BAC-2 and BAC-1. The first is the probability to leave no block state during a step of Markov chain model. The second is the expected time to generate queueing blocks.

Let $\tilde{p}_a$ be the probability that a FN using BAC-2 generates a new block and leaves no block state during a step of Markov chain model. Compared with (1) in BAC-1, the FN using BAC-2 will pause mining whenever a block transmission of the other FN is detected on the channel, so $\tilde{p}_a$ is given by

$$\begin{aligned}\tilde{p}_a=&p_s\cdot 0+p_c\cdot 0+(1-p_s-p_c)\cdot P\{T\le\sigma\}\\ =&(1-p_s-p_c)[1-\exp(-\lambda\sigma)].\end{aligned} \tag{8}$$

Let $\tilde{\alpha}$ be the stationary probability that the queue is not empty after a FN using BAC-2 transmits its block successfully. Let $\tilde{T}_q$ be the expected time to generate queueing blocks in BAC-2. Based on the definition, we have $\tilde{\alpha}=\lambda\tilde{T}_q$. Using BAC-2, a FN performs hash operations when the channel is idle or when the FN transmits its own block. Accordingly, by means of (4), we have

$$\tilde{T}_q=\sum_{i=0}^{m}p_e(i)\left[(iT_c+T_s)+\sum_{n=0}^{i}\frac{W_n-1}{2}\sigma\right]. \tag{9}$$

Then, $\tilde{\alpha}=\lambda\tilde{T}_q$ can be derived. When p_a and α change to $\tilde{p}_a$ and $\tilde{\alpha}$ respectively, the one-step probabilities in (2), (3) and (7) are available for BAC-2.

B. Stationary Probabilities for BAC-1 and BAC-2

Let $\pi_{-1,0}$ denotes the stationary probability of no block state, $\pi_{i,k}$ ($i\in[0,m]$, $k\in[0,W_i-1]$) denote the stationary probabilities of backoff states and transmitting states. Based on the chain regularities of the backoff and transmitting states in Fig. 3, we can obtain

$$\begin{cases} \pi_{i,0}=\frac{p}{W_i}\pi_{i-1,0}+(1-p)\pi_{i,1},\\ \pi_{i,1}=\frac{p}{W_i}\pi_{i-1,0}+(1-p)\pi_{i,2}+p_c\cdot\pi_{i,1},\\ \pi_{i,2}=\frac{p}{W_i}\pi_{i-1,0}+(1-p)\pi_{i,3}+p_c\cdot\pi_{i,2},\\ \cdots,\\ \pi_{i,W_{i-1}}=\frac{p}{W_i}\pi_{i-1,0}+p_c\cdot\pi_{i,W_{i-1}}, \end{cases} \tag{10}$$

where $i\in[1,m]$. Using the second equation of (10), $\pi_{i,0}$ can be rewritten as $\pi_{i,0}=(1+\frac{1-p}{1-p_c})\frac{p}{W_i}\pi_{i-1,0}+\frac{(1-p)^2}{1-p_c}\pi_{i,2}$. In the same way, the other equations in (10) can be used to simplify $\pi_{i,0}$ as follows:

$$\begin{aligned}\pi_{i,0}=&\left[1+\frac{1-p}{1-p_c}+(\frac{1-p}{1-p_c})^2+\cdots+(\frac{1-p}{1-p_c})^{W_{i-1}}\right]\frac{p}{W_i}\pi_{i-1,0}\\ =&\frac{1-[(1-p)/(1-p_c)]^{W_i}}{1-(1-p)/(1-p_c)}\frac{p}{W_i}\pi_{i-1,0},\end{aligned} \tag{11}$$

where $i\in[1,m]$. Based on (11), we obtain

$$\pi_{i,0}=\prod_{n=0}^{i}\frac{1-[(1-p)/(1-p_c)]^{W_n}}{W_n}\left[\frac{p}{1-(1-p)/(1-p_c)}\right]^{i}\pi_{0,0}, \tag{12}$$

where $i\in[1,m]$.

Then, we study the case when $i=0$. From Fig. 3, we can obtain that the regularity to enter stage 0 is

$$\pi_{0,k}=\frac{1}{W_0}\left[p_a\pi_{-1,0}+(1-p)\alpha\sum_{i=0}^{m}\pi_{i,0}\right],\quad k\in[0,W_0-1]. \tag{13}$$

Meanwhile, the regularity to enter stage i ($i\in[1,m]$) is

$$\pi_{i,k}=\frac{1}{W_i}p\pi_{i-1,0},\qquad i\in[1,m],k\in[0,W_i-1]. \tag{14}$$

Based on the regularities in (13) and (14), we can use $p_a\pi_{-1,0}+(1-p)\alpha\sum_{i=0}^{m}\pi_{i,0}$ to replace $p\pi_{i-1,0}$ ($i\in[1,m]$) in (11) and this yields

$$\pi_{0,0}=\frac{1-[(1-p)/(1-p_c)]^{W_0}}{1-(1-p)/(1-p_c)}\frac{p_a}{W_0}\left[\pi_{-1,0}+\frac{(1-p)\alpha}{p_a}\sum_{i=0}^{m}\pi_{i,0}\right]. \tag{15}$$

To simplify (15), we should find the expression of $\pi_{-1,0}$. Based on the chain regularity of no block state in Fig. 3, we have

$$\begin{aligned}\pi_{-1,0}=&(1-p_a)\pi_{-1,0}+(1-p)(1-\alpha)\sum_{i=0}^{m}\pi_{i,0}+p\pi_{m,0}\\&+p_s\left(1-\sum_{i=0}^{m}\pi_{i,0}-\pi_{-1,0}\right).\end{aligned} \tag{16}$$

After simplifying (16), we obtain

$$\begin{aligned}\pi_{-1,0}=&\frac{(1-p)(1-\alpha)}{p_a+p_s}\sum_{i=0}^{m}\pi_{i,0}+\frac{p}{p_a+p_s}\pi_{m,0}\\&+\frac{p_s}{p_a+p_s}\left(1-\sum_{i=0}^{m}\pi_{i,0}\right).\end{aligned} \tag{17}$$

Substituting (12) and (17) into (15) yields

$$\begin{aligned}\pi_{0,0}=&\left[\frac{(1-p)(1-\alpha)}{p_a+p_s}-\frac{p_s}{p_a+p_s}+\frac{(1-p)\alpha}{p_a}\right]\sum_{i=0}^{m}f(i)\pi_{0,0}\\&+\frac{p}{p_a+p_s}f(m)\pi_{0,0}+\frac{p_s}{p_a+p_s}\frac{1-[(1-p)/(1-p_c)]^{W_0}}{1-(1-p)/(1-p_c)}\frac{p_a}{W_0}.\end{aligned} \tag{18}$$

where $f(x)=\prod_{n=0}^{x}\frac{1-[(1-p)/(1-p_c)]^{W_n}}{W_n}\left[\frac{p}{1-(1-p)/(1-p_c)}\right]^{x+1}\frac{p_a}{p}$. After simplifying (18), we obtain the expression of $\pi_{0,0}$ as follows:

$$\pi_{0,0}=\frac{\frac{p_s}{p_a+p_s}\frac{1-[(1-p)/(1-p_c)]^{W_0}}{1-(1-p)/(1-p_c)}\frac{p_a}{W_0}}{1+\left[\frac{p_s}{p_a+p_s}-\frac{(1-p)(1-\alpha)}{p_a+p_s}-\frac{(1-p)\alpha}{p_a}\right]\sum_{i=0}^{m}f(i)-\frac{p}{p_a+p_s}f(m)}. \tag{19}$$

Let τ_1 be the probability that a FN using BAC-1 transmits block in a randomly chosen time slot. It can be expressed as $\tau_1=\sum_{i=0}^{m}\pi_{i,0}$. τ_b rewrites as

$$\tau_1=\left\{1+\sum_{i=1}^{m}\prod_{n=1}^{i}\frac{1-\left[\frac{(1-p)}{(1-p_c)}\right]^{W_n}}{W_n}\left[\frac{p}{1-\frac{(1-p)}{(1-p_c)}}\right]^{i}\right\}\pi_{0,0}, \tag{20}$$

where $\pi_{0,0}$ can be substituted by (19), α can be substituted by (6), p_a can be substituted by (1). The probability p that a collision is seen by a block transmitted on the channel is given by $p=1-(1-\tau_1)^{N-1}$. The probability that the channel contains a successful block transmission of other FNs is given by $p_s=(N-1)\tau_1(1-\tau_1)^{N-2}$. The probability that the channel contains a block collision of other FNs is $p_c=p-p_s$. So τ_1 is the only unknown parameter in (20), which can be obtained through iteration method. Note that (20) can be applied to BAC-2 when p_a and α change to $\tilde{p}_a$ and $\tilde{\alpha}$. To distinguish this difference between BAC-1 and BAC-2, we denote the transmitting probability of BAC-2 by τ_2.

C. Markov Chain Model for BAC-3 and BAC-4

Since the BAC-3 and BAC-4 contain mining strategy II, a FN will pause mining when it enters the block backoff state until the new block is transmitted successfully or discarded. As a result, block queueing does not exist in the Markov chain model in Fig. 4, which is the main difference between the two Markov chain models.

BAC-3: In no block and block backoff states, the expression of the one-step transition probabilities using BAC-3 are the same as BAC-1 in (2) and (3). On the other hand, in block transmitting states, the one-step transition probabilities of BAC-3 are different with BAC-1 in (7). Since without block queueing in BAC-3, a FN must return to no block state after the FN transmits or discards its block. According to the analysis, the one-step transition probabilities of BAC-3 can be expressed as

$$\begin{cases}P\{-1,0\mid -1,0\}=1-p_a,\\ P\{0,k\mid -1,0\}=p_a/W_0, & & k\in[0,W_0-1],\\ P\{-1,0\mid i,k\}=p_s, & i\in[0,m], & k\in[1,W_i-1],\\ P\{i,k\mid i,k\}=p_c, & i\in[0,m], & k\in[1,W_i-1],\\ P\{i,k-1\mid i,k\}=1-p, & i\in[0,m], & k\in[1,W_i-1],\\ P\{i+1,k\mid i,0\}=p/W_{i+1}, & i\in[0,m-1], & k\in[0,W_i-1],\\ P\{-1,0\mid i,0\}=1-p, & i\in[0,m-1], & \\ P\{-1,0\mid m,0\}=1. & & \end{cases} \tag{21}$$

BAC-4: The only difference between the BAC-4 and BAC-3 is the probability to leave no block state. Influenced by mining strategy I, the probability that a FN using BAC-4 leaves no block state during a step of Markov chain model is $\tilde{p}_a=(1-p_s-p_c)[1-\exp(-\lambda\sigma)]$. So when p_a changes to $\tilde{p}_a$, the one-step probabilities in (21) are available for BAC-4.

D. Stationary Probabilities for BAC-3 and BAC-4

Compared the chain regularities in Fig. 3 with that in Fig. 4, we can know that (10) can be directly applied to Fig. 4. Using (10), $\pi_{i,0}$ can be expressed as

$$\pi_{i,0}=\frac{1-[(1-p)/(1-p_c)]^{W_i}}{1-(1-p)/(1-p_c)}\frac{p}{W_i}\pi_{i-1,0}, \tag{22}$$

where $i\in[1,m]$. For $i=0$, we use p_a to replace p in (22) and obtain

$$\pi_{0,0}=\frac{1-[(1-p)/(1-p_c)]^{W_0}}{1-(1-p)/(1-p_c)}\frac{p_a}{W_0}\pi_{-1,0}. \tag{23}$$

Now, using (22) and (23), a equation between no block state and transmitting states can be given by

$$\pi_{i,0} = f(i)\pi_{-1,0}, \tag{24}$$

where $f(x) = \prod_{n=0}^{x} \frac{1-[(1-p)/(1-p_c)]^{W_n}}{W_n} \left[\frac{p}{1-(1-p)/(1-p_c)}\right]^{x+1} \frac{p_a}{p}$ and $i \in [0, m]$. Based on the chain regularity of no block state in Fig. 4, we obtain another equation between no block state and transmitting states as follows:

$$\begin{aligned}\pi_{-1,0} =& (1-p_a)\pi_{-1,0} + (1-p)\sum_{i=0}^{m}\pi_{i,0} + p\cdot\pi_{m,0} \\ &+ p_s\left(1 - \sum_{i=0}^{m}\pi_{i,0} - \pi_{-1,0}\right).\end{aligned} \tag{25}$$

By means of (24), (25) can be rewritten as

$$\pi_{-1,0} = \frac{p_s}{p_a + p_s - (1-p-p_s)\sum_{i=0}^{m} f(i) - pf(m)}. \tag{26}$$

Let the transmitting probability of BAC-3 be $\tau_3 = \sum_{i=0}^{m}\pi_{i,0}$. Using (24) and (26), τ_3 can be expressed as

$$\tau_3 = \frac{p_s \sum_{i=0}^{m} f(i)}{p_a + p_s - (1-p-p_s)\sum_{i=0}^{m} f(i) - pf(m)}, \tag{27}$$

where $p_a = (1-p_s-p_c)[1-\exp(-\lambda\sigma)]$, $p = 1-(1-\tau_3)^{N-1}$, $p_s = (N-1)\tau_3(1-\tau_3)^{N-2}$, and $p_c = p - p_s$. So τ_3 is the only unknown parameter in (27), which can be obtained through iteration method. Note that (27) can be applied to BAC-4 when p_a changes to $\tilde{p}_a$. To distinguish this difference between BAC-3 and BAC-4, we denote the transmitting probability of BAC-4 by τ_4.

V. Performance Analysis

In the section, based on the stationary probabilities of Markov chain models, we analyse the closed-form expressions of the key performance metrics in B-WLAN by involving the impact of four BAC approaches.

A. Transaction Throughput

Transaction throughput is defined as the number of transactions that included by a valid block per second, i.e., transaction per second (tps). We denote transaction throughput by θ_t. To calculating the θ_t, we can multiply the number of blocks that successfully transmitted on the channel per second by the maximum number of transactions in a block. The maximum number of transactions in a block relates to the block size. To reduce the forking probability and achieve consensus in B-WLAN, we consider that a FN can transmit a full block after the backoff counter, which contains a block header and all related transactions. Let s_b be the size of a block, s_h be the size of the block header, s_t be the average size of a transaction, N_t be the maximum number of transactions in a block. Since each transmission contains a block header and

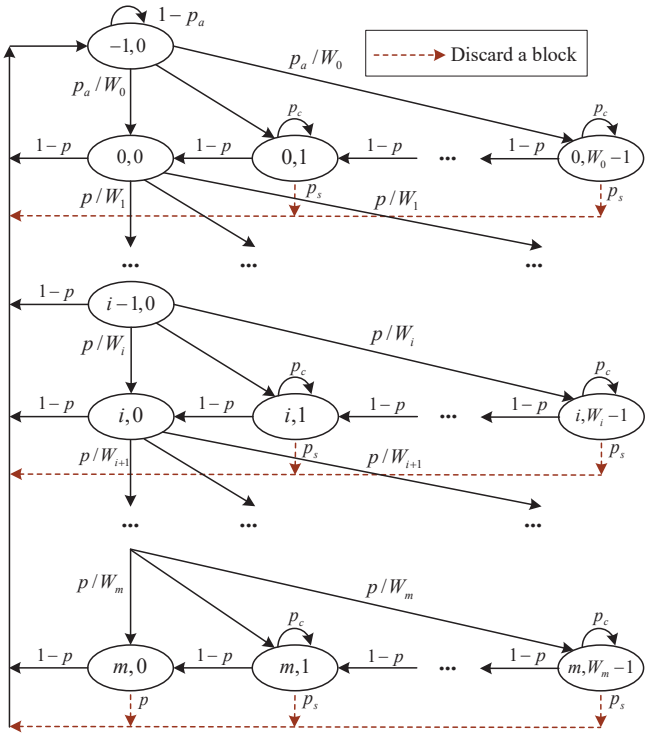


Fig. 4: Markov chain model for BAC-3; When p_a changes to $\tilde{p}_a$, this model is available for BAC-4.

all related transactions, we have $N_t = (s_b - s_h)/s_t$. Based on the throughput analysis in [19], the transaction throughput θ_t can be given by

$$\theta_t = \frac{p_1 N_t}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c}, \tag{28}$$

where $p_0 = (1-\tau)^N$, $\tau \in \{\tau_1, \tau_2, \tau_3, \tau_4\}$ representing the probability that the channel stays idle in a slot. $p_1 = N\tau(1-\tau)^{N-1}$, $\tau \in \{\tau_1, \tau_2, \tau_3, \tau_4\}$ representing the probability that a successful transmission occurs. Accordingly, $1 - p_0 - p_1$ is the probability that a collision happens. T_s is the average channel busy time when a successful transmission occurs. T_c is the average channel busy time when a collision happens. According to [19], T_s and T_c can be given by

$$\begin{cases} T_s = H + s_b + \text{SIFS} + \delta + \text{ACK} + \text{DIFS} + \delta, \\ T_c = H + s_b + \text{DIFS} + \delta, \end{cases} \tag{29}$$

where H is the packet header; δ is the propagation delay; SIFS, DIFS and ACK denote the frame duration.

Note that by substituting $\tau \in \{\tau_1, \tau_2, \tau_3, \tau_4\}$ into p_0 and p_1, the transaction throughput of BAC-1, BAC-2, BAC-3 and BAC-4 can be derived respectively using (28). For example, when $p_0 = (1-\tau_1)^N$ and $p_1 = N\tau_1(1-\tau_1)^{N-1}$, the result of (28) is the transaction throughput of BAC-1.

B. Block Discard Rate

Based on mining strategy, the way to derive block discard rate is different among four BAC approaches.

BAC-1: Let θ_{d1} be the block discard rate of BAC-1 representing the number of forking blocks discarded by all FNs per second. Since BAC-1 do not contain mining strategy to pause

mining, the FNs will keep mining all the time. In this case, θ_{d1} can be expressed as

$$\theta_{d1} = \lambda N - \theta_s, \tag{30}$$

where λN is the block generation rate of whole network. θ_s is the block successful transmission rate of whole network representing the number of blocks successfully transmitted by all FNs per second, which can be given by

$$\theta_s = \frac{p_1}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c}. \tag{31}$$

BAC-2: Let θ_{d2} be the block discard rate of BAC-2. According to mining strategy I, the FNs using BAC-2 should perform hash operations based on the channel condition. When the channel stays idle (probability p_0), all FNs will perform hash operations. When the channel contains a successful block transmission (probability p_1), only the FN in block transmitting states will perform hash operations. When the channel contains a collision (probability $1-p_0-p_1$), all the FNs in block transmitting states will perform hash operations. Based on the analysis, θ_{d2} can be given by

$$\theta_{d2} = \frac{p_0 N\lambda\sigma + p_1\lambda T_s + \sum_{j=2}^{N}\binom{N}{j}\tau^j(1-\tau)^{N-j} j\lambda T_c}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c} - \theta_s, \tag{32}$$

where j is the expected number of FNs in block transmitting states during a collision, and the collision probability $1-p_0-p_1 = \sum_{j=2}^{N}\binom{N}{j}\tau^j(1-\tau)^{N-j}$.

BAC-3: Let θ_{d3} be the block discard rate of BAC-3. Instead of basing on channel condition, mining strategy II pauses mining based on the working states of a FN. So we use another way to study the block discard rate of BAC-3. The blocks will be discarded because of the following two cases. (i) When a successful transmission occurs on the channel, the FNs in no block state or block backoff states will discard blocks based on discard strategy. (ii) When a collision happens on the channel, the FNs in state $\{m, 0\}$ will discard blocks based on CSMA/CA.

We first analyse the case (i). In B-WLAN, a successful transmission occurs with probability $p_1 = N\tau(1-\tau)^{N-1}$, where τ is the transmitting probability and N is the number of FNs. Based on this expression, we can know that one of the FNs is in transmitting state and the other $N-1$ FNs are in no block state or block backoff states in this slot. The number of discarded blocks depends on how many FNs are in no block state and block backoff states. Using Binomial theorem [20], $(1-\tau)^{N-1}$ can be expanded as

$$(1-\tau)^{N-1} = \sum_{n_b=0}^{N-1}\binom{N-1}{n_b}{\pi_{-1,0}}^{N-1-n_b}\cdot(1-\tau-\pi_{-1,0})^{n_b}, \tag{33}$$

where n_b is the number of FNs in block backoff states, $N-1-n_b$ is the number of FNs in no block state. Accordingly, we can know that the number of discarded blocks when a successful block transmission occurs is

$$n_s = n_b + (N-1-n_b)[1-\exp(-\lambda T_s)]. \tag{34}$$

Using (33) and (34) to calculate expected value, the block discard rate for case (i) can be given by

$$\theta_{ds} = \frac{\sum_{n_b=0}^{N-1} n_s \cdot N\tau\binom{N-1}{n_b}{\pi_{-1,0}}^{N-1-n_b}\cdot(1-\tau-\pi_{-1,0})^{n_b}}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c}. \tag{35}$$

Now we analyse the case (ii). In B-WLAN, a collision happens with probability $1-p_0-p_1 = \sum_{j=2}^{N}\binom{N}{j}\tau^j(1-\tau)^{N-j}$. Based on this expression, we can know that j FNs are in transmitting state and $N-j$ FNs are in no block state or block backoff states in this slot. The number of discarded blocks depends on how many FNs are in state $\{m, 0\}$. So we use Binomial theorem to expand τ^j as

$$\tau^j = \sum_{n_c=0}^{j}\binom{j}{n_c}(\tau-\pi_{m,0})^{j-n_c}\cdot{\pi_{m,0}}^{n_c}, \tag{36}$$

where $j \in [2, N]$. n_c is the number of FNs in state $\{m, 0\}$, and thus the number of discarded blocks when a collision happens is equal to n_c. Based on this analysis, the block discard rate for case (ii) can be given by

$$\theta_{dc} = \frac{\sum_{n_c=0}^{j} n_c \cdot \sum_{j=2}^{N}\binom{N}{j}(1-\tau)^{N-j}\binom{j}{n_c}(\tau-\pi_{m,0})^{j-n_c}\cdot{\pi_{m,0}}^{n_c}}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c}. \tag{37}$$

Using (35) and (37), the block discard rate of BAC-3 is given by

$$\theta_{d3} = \theta_{ds} + \theta_{dc}. \tag{38}$$

BAC-4: Let θ_{d4} be the block discard rate of BAC-4. The difference between BAC-4 and BAC-3 is that the FN in no block state will pause mining during the block transmission, and thus the FN only performs hash operations during the channel idle time. So when a successful block transmission occurs in BAC-4, the FNs in no block state will not discard blocks. In this case, $n_s = n_b$ for BAC-4, and we rewrite (35) as

$$\tilde{\theta}_{ds} = \frac{\sum_{n_b=0}^{N-1} n_b \cdot N\tau\binom{N-1}{n_b}{\pi_{-1,0}}^{N-1-n_b}\cdot(1-\tau-\pi_{-1,0})^{n_b}}{p_0\sigma + p_1 T_s + (1-p_0-p_1)T_c}. \tag{39}$$

Using (39) and (37), the block discard rate of BAC-4 is given by

$$\theta_{d4} = \tilde{\theta}_{ds} + \theta_{dc}. \tag{40}$$

C. Block Utilization and Mining Pause Probability

In blockchain network, when a block is transmitted successfully, it will be verified and stored by all FNs. The computational power included in this block will be used to enhance the security of the public ledger. To study how many blocks can be used for security, let η be the block utilization in

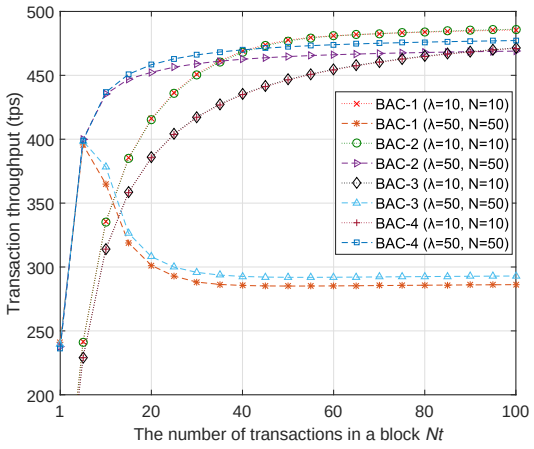


Fig. 5: Transaction throughput vs. N_t

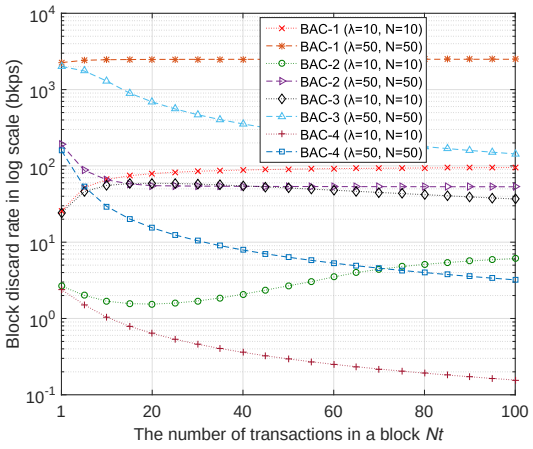


Fig. 6: Block discard rate (log scale) vs. N_t

B-WLAN, defined as the (long-run) proportion of the blocks that is transmitted successfully.

Using block successful transmission rate θ_s, the number of blocks successfully transmitted by FNs during the time period T is $\theta_s T$. Using block discard rate $\theta_d \in \{\theta_1, \theta_2, \theta_3, \theta_4\}$, we can obtain that the number of blocks discarded by FNs during the time period T is $\theta_d T$. So the block utilization is given by

$$\eta = \lim_{T\to\infty} \frac{\theta_s T}{\theta_s T + \theta_d T} = \frac{\theta_s}{\theta_s + \theta_d}, \tag{41}$$

where θ_s can be derived by (31). $\theta_d \in \{\theta_1, \theta_2, \theta_3, \theta_4\}$ can be derived by (30), (32), (38) and (40), respectively.

Now, we study the stationary mining pause probability, which equals the (long-run) proportion of time that the mining is paused. Let this probability be p_m. Review that the average number of blocks generated by a FN per second is $\lambda = r/D$, where r is the hashrate of a FN and D is the hash difficulty. Without mining strategy, the whole network will generate average λNT blocks during the time period T. But in fact, because of the mining pause, the number of blocks generated by the whole network during T decreases to $(\theta_s + \theta_d)T$. This yields

$$p_m = \lim_{T\to\infty} \frac{\lambda NT - (\theta_s + \theta_d)T}{\lambda NT} = \frac{\lambda N - \theta_s - \theta_d}{\lambda N}. \tag{42}$$

VI. Numerical Results and Discussions

In this section, we use Matlab to calculate the closed-form expression of the performance metrics for comparing the performance of B-WLAN using BAC-1, BAC-2, BAC-3 and BAC-4, respectively.

A. Parameter Settings

We consider a B-WLAN with N FNs. Based on [19], the minimum contention window W_{min} is set as 16, the maximum contention window W_{max} is set as 1024, the maximum backoff stage m is set as 6, the packet header H is set as 400 bits, the size of ACK frame is set as 240 bits, channel bit rate is set as 1 Mbit/s, the propagation delay δ is set as 1 μs, the time slot σ is set as 50 μs. Meanwhile, based on [14], the size of the block header s_h is set as 640 bits, the size of a transaction s_t is set as 2000 bits.

B. Performance Evaluation

In the first experiment, we vary the number of transactions in a block N_t from 1 to 100 to study the impact of N_t on transaction throughput, block discard rate, block utilization and mining pause probability. Meanwhile, we set the number of FNs N and the block generation rate λ as 10 and 50 to conduct comparisons. The unit of λ is block per second (bkps).

The transaction throughput of BAC-1, BAC-2, BAC-3 and BAC-4 are derived by substituting the stationary probabilities τ_1, τ_2, τ_3 and τ_4 into (28) respectively. **BAC-1**: Fig. 5 shows that when $\lambda = 10$, $N = 10$, the BAC-1 can help B-WLAN to achieve a high transaction throughput, i.e., up to 480 tps in ideal channel condition. With the increase of N_t, one block can contain more transactions after the backoff counter and the backoff delay of a single transaction decreases, so the transaction throughput quickly increases at the beginning. But at the same time, due to limited channel resource, the increasing rate declines gradually, and the throughput reaches a upper bound. When $\lambda = 50$ and $N = 50$, the transaction throughput of BAC-1 begin to decrease after $N_t = 5$ and cannot reach the upper bound. Because when λ and N is large, there are too many blocks occurring in the backoff states and the collisions cannot be effectively avoided by the backoff counter. **BAC-2**: Compared with BAC-1, BAC-2 can help B-WLAN to reach the throughput upper bound, no matter $\lambda = 10$, $N = 10$ or $\lambda = 50$, $N = 50$. This phenomenon reflects that mining strategy I can effectively pause mining to balance the block generation rate. This action helps the backoff scheme to avoid collisions and maintain a high transaction throughput in high load case. **BAC-3**: The curves under BAC-3 is similar to that under BAC-1, while BAC-1 behaves sightly better in small λ and N, and BAC-3 behaves better in large λ and N. This means strategy II is suitable for high block generation rate case. **BAC-4**: It is shown that BAC-4 has the highest throughput when $\lambda = 50$, $N = 50$. So BAC-4 is the best choice in this high load case. For $\lambda = 10$, $N = 10$, BAC-1 and BAC-2 have a very similar curve, but BAC-2 is a better choice in low load case since the mining strategy in it saves power.

The block discard rate of BAC-1, BAC-2, BAC-3 and BAC-4 are derived by substituting τ_1, τ_2, τ_3 and τ_4 into (30), (32), (38) and (40) respectively. **BAC-1**: Fig. 6 shows that

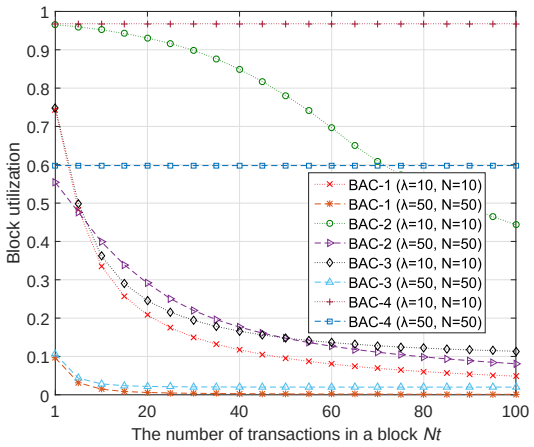


Fig. 7: Block utilization vs. N_t

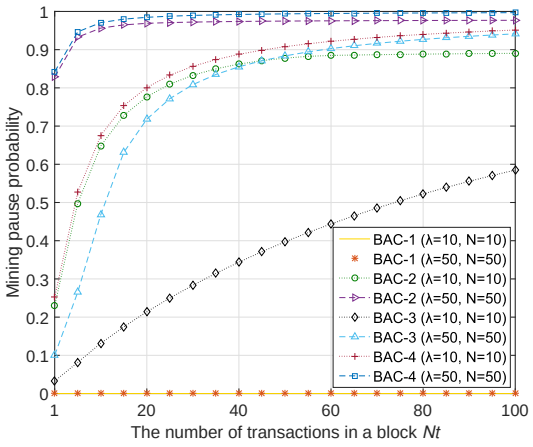


Fig. 8: Mining pause probability vs. N_t

when $\lambda = 10$, $N = 10$ and $\lambda = 50$, $N = 50$, the block discard rate of BAC-1 always increases with N_t. Because a larger N_t means a longer block transmission time, and BAC-1 does not pause mining during the transmission time. So more forking blocks are generated and discarded. **BAC-2**: It can be noticed that when $\lambda = 10$, $N = 10$ the block discard rate firstly decreases and then increases with N_t. The reason is that with the increase of N_t, the block transmission time becomes longer, which increases the mining pause time and reduces the forking probability. So the discard rate decreases with N_t at first. But on the other hand, a longer block transmission time also incurs more queuing blocks. A large number of queuing blocks cause a high forking probability and cannot be well addressed by mining pause. So the discard rate increases finally. When $\lambda = 50$, $N = 50$, the block discard rate of BAC-2 always decrease with N_t. This is because a large $\lambda = 50$ and $N = 50$ results in much collision on the channel, and mining strategy I frequently pause mining to reduces the forking blocks. **BAC-3**: It is shown that when $\lambda = 10$, $N = 10$ the block discard rate firstly increases and then decreases with N_t. Because the FN using BAC-3 keeps mining during the block transmission of other FNs, more forking blocks are discarded after a block transmission. But on the other hand, with a much longer transmission time, the impact of mining pause significantly increases, which slows down the total block generation rate and reduces forking probability. For $\lambda = 50$, $N = 50$, the block discard rate always decrease with N_t due to high block generation rate prolonging the average mining pause time. **BAC-4**: The block discard rate always decreases with N_t. For a given λ and N, the block discard rate of BAC-4 is lower than the other BAC approaches. This means mining strategy I and strategy II are both effective to reduce the block discard rate.

The block utilization of BAC-1, BAC-2, BAC-3 and BAC-4 are derived by substituting τ_1, τ_2, τ_3 and τ_4 into (41) respectively. **BAC-1**: Fig. 7 shows that the block utilization of BAC-1 always decease with N_t. Because a large N_t prolongs block transmission time, and much computational power of FNs is wasted before receiving the new block, which results in a low block utilization. **BAC-2**: The block utilization also decease with N_t. But for a given N_t, the block utilization of BAC-2 is higher than BAC-1, since the mining strategy I in it reduces forking probability. **BAC-3**: For a given N_t, it is shown that the block utilization of BAC-3 is lower than that of BAC-2. This phenomenon reflects that mining strategy I has a better effect on block utilization, compared with strategy II. The reason is that strategy I pauses mining based on the detection of block transmission which happens more frequently than strategy II. But on the other hand, the implement cost of strategy I will be higher than strategy II. **BAC-4**: It is observed that the block utilization of BAC-4 is not affected by N_t and stabilize at a high level. This important results indicates that two mining strategies can work well together to achieve a optimal block utilization.

The mining pause probability of BAC-1, BAC-2, BAC-3 and BAC-4 are derived by substituting τ_1, τ_2, τ_3 and τ_4 into (42) respectively. **BAC-1**: BAC-1 does not contain mining strategy, so the mining pause probability is 0. **BAC-2**: Fig. 8 shows that the mining pause probability of BAC-2 increases with N_t. Because a larger N_t needs a longer transmission time, it increases mining pause time. **BAC-3**: The mining pause probability of BAC-3 is lower than that of BAC-2. This is because strategy I pauses mining whenever a block is transmitted, and mining strategy II pauses mining when a FN transmit its own block. For example, when the channel contains a block transmission, there are $N - 1$ FNs pausing mining in BAC-2, but there is only one FN pausing mining in BAC-3. **BAC-4**: For a given N_t, BAC-4 has a highest mining pause probability, which means it saves more computational power than other BAC approaches.

In the second experiment, we vary the block generation rate λ at each FN from 1 to 100 to compare the performance of four BAC approaches with the number of FNs. Meanwhile, we set $N_t = 10$ in this experiment, and one can capture the impact of N_t based on the result of first experiment.

BAC-1: Fig. 9 shows that the transaction throughput of BAC-1 increases with block generation rate λ before reaching the maximum value. Because when λ is low there are very few blocks transmitting on the channel, the collision probability is low and the transaction throughput increases quickly with λ. With the increases of λ, the collisions are addressed by the CSMA/CA scheme and thus the transaction throughput reach-

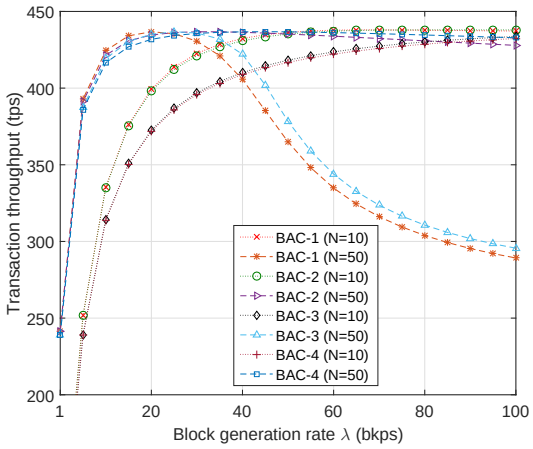


Fig. 9: Transaction throughput vs. λ

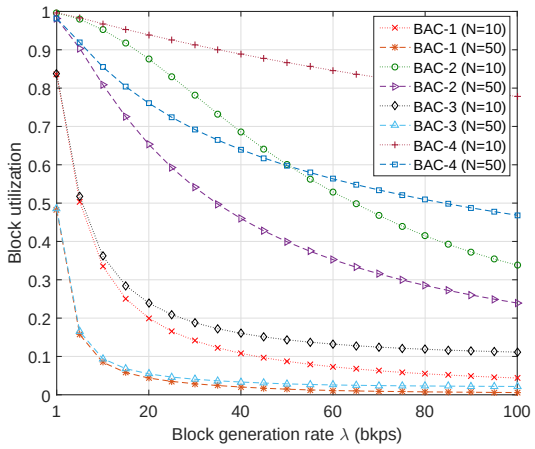


Fig. 11: Block utilization vs. λ

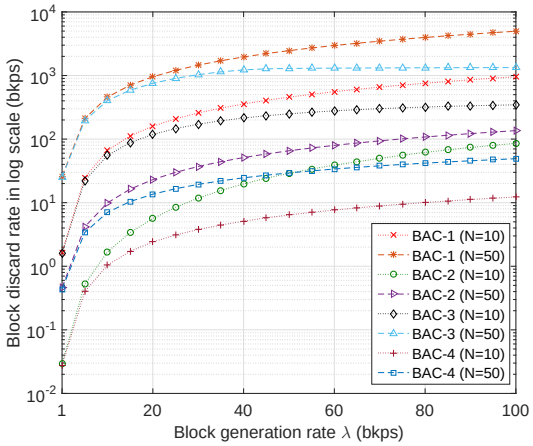


Fig. 10: Block discard rate (log scale) vs. λ

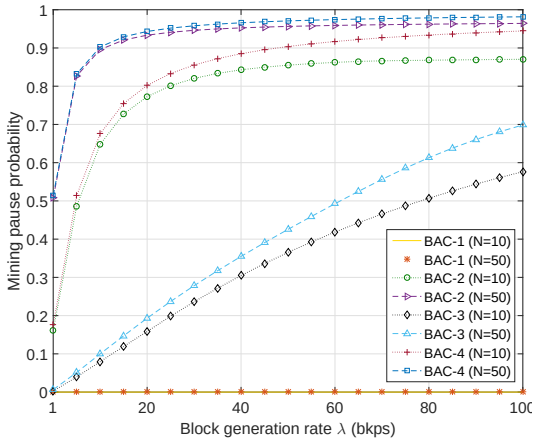


Fig. 12: Mining pause probability vs. λ

es the maximum value. After that, the transaction throughput decreases with λ due to too many blocks occurring in the backoff states and the collisions cannot be effectively avoided by the backoff counter. **BAC-2**: When $N = 10$, the BAC-2 has a similar throughput with BAC-1. However, BAC-1 achieves this throughput by using more computational power, while BAC-2 achieves this throughput with a higher implement cost (block detection). When $N = 50$, the transaction throughput of BAC-2 decreases much slower than BAC-1 after reaching the maximum value. Because BAC-2 contains mining strategy I to pause mining and slow down λ in high load case, which reduces collision probability and maintains a high throughput. **BAC-3**: BAC-3 is suitable for the case when four BAC approaches have a similar throughput, e.g., $\lambda = 100$, $N = 10$ or $\lambda = 20$, $N = 50$. This is because the implement of mining strategy II is much easier than strategy I (based on the principles in section III). So for implement cost, BAC-1 is similar to BAC-3, which are both lower than BAC-2 and BAC-4. Meanwhile, the power consumption of BAC-1 is higher than BAC-3. **BAC-4**: It is shown that the curve of BAC-4 is similar to BAC-2 when $N = 50$, while BAC-4 has a lower power consumption. So BAC-4 is a better choice for an optimal throughput with large λ and N, e.g., $\lambda = 100$ and $N = 50$.

Fig. 10 shows that the block discard rate of four BAC approaches increases with λ. Because a larger λ results in more blocks simultaneously generated in the backoff state. In this case, only one block can be transmitted successfully, and the other blocks are discarded due to forking. So the block discard rate increases with λ. Meanwhile, it can be observed that the block discard rate of BAC-1 is higher than the other BAC approaches. This means mining strategy can reduce forking blocks effectively, especially in the scenario with high block generation rate.

Fig. 11 shows that the block utilization of four BAC approaches always decrease with λ. When block utilization is lower than 0.5, most blocks generated by FNs are discarded due to forking. The reason is that when multiple blocks simultaneously generated in the backoff states, the discard probability for a single block is very high. Since the forking blocks cannot protect the main chain of the ledger, a low block utilization "dilutes" the computational power of FN and affects the security of blockchain network. In addition, Fig. 9 and Fig. 11 demonstrate the trade-off between transaction throughput and block utilization. This means one can adjust PoW hash difficulty to accelerate block generation rate for achieving a high transaction throughput, but at the same time the block utilization decreases.

Fig. 12 shows that the mining pause probability of BAC-2, BAC-3 and BAC-4 always increase with λ. Meanwhile, for a given N, it is shown that the mining pause probability of BAC-4 is the highest, and the mining pause probability of BAC-

2 is higher than that of BAC-3. Note that the mining pause probability affects the computational power consumption of FNs. So the results in Fig. 11 and Fig. 12 prove that mining strategy can save the computational power while improving block utilization of B-WLAN.

Note that the performance comparisons in this work refer to four BAC approaches that contain different strategies. For the baseline approach without any strategies, it can be predicted that the performance would be worse than BAC-1, since the forking blocks in queue consume much channel resource and increase block propagation delay.

VII. Related Work

In recent years, many researches have been carried out to improve the transaction throughput of PoW-based blockchain. Bitcoin-NG [22] selects a leader to post multiple blocks, thus increasing the block generation rate of PoW for improving the transaction throughput. Hybrid-IoT [23] proposes a two-tier blockchain architecture, where subgroups of full nodes achieve consensus through PoW algorithm and the connection among the sub-blockchains employs a Byzantine fault-tolerant framework. Monoxide [24] adopts multiple independent and parallel PoW sub-blockchains termed as zones, in which different zones can conduct trading using the cross-zone algorithm. Tangle [25] uses a directed acyclic graph (DAG)-based ledger to replace the conventional single chain-based ledger. After solving a simple PoW task, the nodes can insert their blocks into DAG-based ledger at any time, which result in a much higher throughput. Although the high throughput can be achieved by these new blockchain systems, the security is compromised since generating multiple sub-blockchains "dilutes" the mining power of honest nodes. Without generating sub-blockchains, this paper proposes a discard strategy to address the blockchain forking problem, which act as a key enabler to accelerate block generation rate and improve transaction throughput. The discard strategy is proposed by considering the impact of CSMA/CA channel contention on blockchain consensus process, and has not been studied in previous work.

The high computational power consumption is also a serious problem in PoW-based blockchain. To address this problem, the authors in [26] propose an auction-based market model to offload the PoW computational tasks to the cloud/fog computing server. Two bidding schemes, i.e, constant-demand scheme and multi-demand scheme are considered to maximize the social welfare of the blockchain network. In [27], the authors study the offloading from miners to cloud/edge servers using a multi-leader multi-follower game. Based on alternating direction method of multipliers (ADMM) algorithm, the utilities of miners and the profits of servers are jointly optimized. Considering the game theory cannot deal with the dynamics of the wireless environment, the authors in [28] propose a multi-agent deep reinforcement learning (DRL) approach to minimize the long-term cost of the PoW task offloading. The above work addresses power consumption of PoW based on offloading model. However, PoW task offloading cannot save the overall computational power, since the server becomes a substitute for miner. Different from offloading approaches, we study the relationship between block transmission delay and forking problem, then propose mining strategy to pause mining during block transmission, which reduces the meaningless computational power consumption on forking blocks.

To validate the effectiveness of the proposed approach, mathematical models are required to quantitatively study the performance of blockchain. In [29], the authors present Markov chain model to analyse the consensus process of DAG-based ledger. Using transition probabilities, the authors derive blockchain performance metrics, i.e., cumulative weight increasing rate and consensus delay. In [30], the authors uses Markov chain model to analyse the performance of Raft consensus algorithm. The impact of packet loss rate, election timeout, and network size on Raft network split probability have been derived. Except Markov chain models, the authors in [31] employ a signal-to-interference-plus-noise ratio (SINR) model to analyse the transaction transmission successful rate and transaction throughput of PoW-based blockchain. An optimal node deployment algorithm has been designed to maximize transaction throughput. However, the above models do not study the impact of communication protocol (e.g. CSMA/CA) on blockchain performance. In [32], the authors develop a queuing model to analyze the impact of CSMA/CA-based transmission delay on the consensus efficiency of DAG-based ledger. But the CSMA/CA-based transmission delay is considered as an average value to increase confirmation delay, and the authors do not study how the random backoff scheme in CSMA/CA affects the block transmission. In view of this, this paper extends the Markov chain model in [19] to study how the proposed strategies and random backoff scheme affect the performance of blockchain. To the best of our knowledge, the transaction throughput has not been studied by involving blockchain strategy and the channel contention of CSMA/CA simultaneously. Meanwhile, the block discard rate, block utilization and mining pause probability are the key performance metrics to show the effectiveness of the proposed strategies, which has not been analysed in previous work.

VIII. Conclusions and Future work

In this work, we propose mining strategies and a discard strategy to reduce the meaningless computational power consumption on forking blocks and improve the transaction throughput in B-WLAN. Based on the proposed strategies, we design four BAC approaches and use Markov chain models to conduct the performance comparisons. Calculation results show that the discard strategy in BAC can help B-WLAN to achieve a high transaction throughput, which could meet the needs of the massive service requests in next-generation wireless network. Meanwhile, the mining strategy can pause the mining of FN to reduce forking probability effectively, which both saving the computational power and improving block utilization. In addition, it is shown that the block size (related to the number of transactions in a block) and PoW hash difficulty (related to block generation rate) will affect the trade-off between transaction throughput and block utilization, which can provide an analytical guideline for building a optimal and secure B-WLAN in the future.

As future work, we will study the performance of baseline approach without any strategies. The analysis of forking probability in queuing and backoff process is the main challenge to derive the transaction throughput and block utilization.

APPENDIX
PROOF OF T_q IN EQUATION (4)

T_q is defined as the expected time of a block spent on backoff and transmitting states counting from the block generation to the successful transmission, which can be given by

$$T_q = \sum_{i=0}^{m} p_e(i) \left[(iT_c + T_s) + \sum_{n=0}^{i} \frac{W_n - 1}{2} \left(\sigma + \frac{p_c}{1-p} T_c \right) \right]. \quad (43)$$

Proof: According to the definition of T_q, we have

$$T_q = T_b + T_t, \quad (44)$$

where T_b and T_t are defined as the expected time spent on backoff states and transmitting states respectively.

Now we analyse T_b and T_t. After a block enters the exponential backoff scheme, it can exit the backoff scheme either through a successful transmission in block transmitting states $\{i, 0\}$ ($i \in [0, m]$) or through a block discard (the dotted lines in Fig. 3). If the block exits the backoff scheme through a block discard, the queue of the FN will be cleared and thus the time spent on backoff and transmitting states should be considered as 0 when we calculate the expected value. Therefore, to derive T_b and T_t, we can only calculate the case when the block exits the backoff scheme through a successful transmission in transmitting states. Let $p_e(i)$ ($i \in [0, m]$) be the probability that a block exits the backoff scheme through a successful transmission in state $\{i, 0\}$. In any backoff state, the probability that a block is not discarded can be given by

$$(1-p) + (1-p)p_c + (1-p)p_c^2 + \cdots + (1-p)p_c^{\infty} = \frac{1-p}{1-p_c}. \quad (45)$$

Using (45), the probability that a block is not discarded during backoff stage i ($i \in [0, m]$) can be expressed as

$$\begin{aligned} &\frac{1}{W_i}\left(\frac{1-p}{1-p_c}\right)^0 + \frac{1}{W_i}\left(\frac{1-p}{1-p_c}\right)^1 + \cdots + \frac{1}{W_i}\left(\frac{1-p}{1-p_c}\right)^{W_i - 1} \\ &= \frac{1}{W_i} \frac{1 - [(1-p)/(1-p_c)]^{W_i}}{1 - (1-p)/(1-p_c)}. \end{aligned} \quad (46)$$

By means of (46), the probability that a block exits the backoff scheme through a successful transmission in state $\{i, 0\}$ is

$$p_e(i) = \prod_{n=0}^{i} \frac{1 - [(1-p)/(1-p_c)]^{W_n}}{W_n} \left[\frac{p}{1 - (1-p)/(1-p_c)} \right]^{i+1} \frac{1-p}{p}. \quad (47)$$

So we obtain the expected time spent on transmitting states is

$$\begin{aligned} T_t &= p_e(0)T_s + p_e(1)(T_c + T_s) + \cdots + p_e(m)(mT_c + T_s) \\ &= \sum_{i=0}^{m} p_e(i)(iT_c + T_s). \end{aligned} \quad (48)$$

For T_b, there are too many possible paths to go through backoff states in Fig. 3, and thus it is very difficult to directly calculate the expected time spent on backoff states. Therefore, we use average value to estimate T_b. Let $\overline{W}$ be the average number of backoff windows, and T_w be the average time spent on each backoff window. By estimation, T_b is given by

$$T_b \approx \overline{W} \cdot T_w, \quad (49)$$

where $\overline{W}$ can be given by

$$\begin{aligned} \overline{W} &= p_e(0)\frac{W_0 - 1}{2} + p_e(1)\sum_{n=0}^{1} \frac{W_n - 1}{2} + \cdots + p_e(m)\sum_{n=0}^{m} \frac{W_n - 1}{2} \\ &= \sum_{i=0}^{m} \sum_{n=0}^{i} p_e(i) \frac{W_n - 1}{2}, \end{aligned} \quad (50)$$

and T_w can be expressed as

$$T_w = \sigma + \frac{p_c}{1-p} T_c. \quad (51)$$

By means of (50) and (51), (49) rewrites as

$$T_b \approx \sum_{i=0}^{m} \sum_{n=0}^{i} p_e(i) \frac{W_n - 1}{2} \left(\sigma + \frac{p_c}{1-p} T_c \right). \quad (52)$$

After we have T_t in (48) and T_b in (52), T_q can be expressed as (43).

REFERENCES

[1] Z. Xiong, Y. Zhang, and *et al.*, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33-39, Aug. 2018.

[2] M. Liu, F. R. Yu, and *et al.*, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695-708, Jan. 2019.

[3] S. R. Pokhrel, J. Choi, and *et al.*, "Federated learning with blockchain for autonomous vehicles: analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734-4746, Aug. 2020.

[4] M. Cebe, E. Erdin, and *et al.*, "Block4forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 50-57, Oct. 2018.

[5] B. Cao, Y. Li, and *et al.*, "When Internet of Things meets blockchain: challenges in distributed consensus," *IEEE Netw.*, vol. 33, no. 6, pp. 133-139, Nov.-Dec. 2019.

[6] Y. Zhang, S. Kasahara,, and *et al.*, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1594-1605, Jun. 2018.

[7] J. Wang, N. Lu, and *et al.*, "A secure spectrum auction scheme without the trusted party based on the smart contract," *Digit. Commun. Netw.*, July 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235286481930330X.

[8] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," White paper, 2009. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[9] G. BitFury, "Proof of stake versus proof of work," White paper, Sep. 2015. [Online]. Available: https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf.

[10] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," In *Proc. Symp. Oper. Syst. Design Implement.*, New Orleans, LA, USA, 1999.

[11] H. Xu, L. Zhang, and *et al.*, "RAFT based wireless blockchain networks in the presence of malicious jamming," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 6, pp. 817-821, Jun. 2020.

[12] T. Salman, M. Zolanvari, and *et al.*, "Security services using blockchains: a state of the art survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 858-880, First Quarter 2019.

[13] J. Xie, F. R. Yu, and *et al.*, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166-173, Sept.-Oct. 2019.

[14] A. M. Antonopoulos, "Mastering Bitcoin: unlocking digital cryptocurrencies," 2nd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., June 2017.

[15] M. Rosenfeld, "Analysis of hashrate-based double-spending," 2014. [Online]. Available: https://arxiv.org/pdf/1402.2009.pdf

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

[17] J. Wan, J. Li, and *et al.*, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3652-3660, Jun. 2019.

[18] B. Bellalta, "IEEE 802.11ax: high-efficiency WLANs," *IEEE Wirel. Commun.*, vol. 23, no. 1, pp. 38-46, Feb. 2016.

[19] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[20] S. Ross, "A first course in probability," 9th ed. New Jersey, USA: Pearson Education, Inc., 2012.

[21] S. M. Ross, "Introduction to probability models,"Academic Press, 2014. $11th$ edition.

[22] I. Eyal, A. E. Gencer, and *et al.*, "Bitcoin-NG: a scalable blockchain protocol," In *Proc. USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, USA, Mar. 2016.

[23] G. Sagirlar, B. Carminati, and *et al.*, "Hybrid-IoT: hybrid blockchain architecture for Internet of Things-pow sub-blockchains," In *Proc. IEEE iThings. GreenCom. CPSCom. SmartData.*, 2018.

[24] J. Wang, and H. Wang, "Monoxide: scale out blockchains with asynchronous consensus zones," In *Proc. USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, USA, Feb. 2019.

[25] S. Popov, "The tangle," White paper, 2018. [Online]. Available: https://www.iota.org/research/academic-papers.

[26] Z. Xiong, S. Feng, and *et al.*, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 4585-4600, Jun. 2019.

[27] Z. Xiong, J. Kang, and *et al.*, "Cloud/edge computing service management in blockchain networks: multi-leader multi-follower game-based ADMM for pricing," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 356-367, Mar.-Apr. 2020.

[28] Z. Li, M. Xu, and *et al.*, "NOMA-enabled cooperative computation offloading for blockchain-empowered Internet of Things: a learning approach," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2364-2378, Feb. 2021.

[29] Y. Li, B. Cao, and *et al.*, "Direct acyclic graph-based ledger for Internet of Things: performance and security analysis," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1643-1656, Aug. 2020.

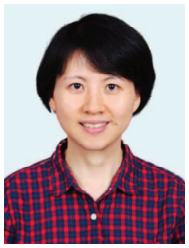
[30] D. Huang, X. Ma, and *et al.*, "Performance analysis of the Raft consensus algorithm for private blockchains," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 1, pp. 172-181, Jan. 2020.

[31] Y. Sun, L. Zhang, and *et al.*, "Blockchain-enabled wireless Internet of Things: performance analysis and optimal communication node deployment," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 5791-5802, Mar. 2019.

[32] B. Cao, M. Li, and *et al.*, "How does CSMA/CA affect the performance and security in wireless blockchain networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4270-4280, Jun. 2020.

YIXIN LI received the M.E degree in information and communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2020. He currently is pursuing his Ph.D. degree at the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China. His research interests include blockchain and wireless communication.

BIN CAO is currently an associate professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Before that, he was an associate professor at Chongqing University of Posts and Telecommunications. He received his Ph.D. degree (Honors) in communication and information systems from the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China in 2014. From April to December in 2012, he was an international visitor at the Institute for Infocomm Research (I2R), Singapore. He was a research fellow at the National University of Singapore from July 2015 to July 2016. He also served as symposium cochair for IEEE ICNC 2018, workshop cochair for CyberC 2019 and TPC member for numerous conferences. His research interests include blockchain system, internet of things and mobile edge computing.



LIANG LIANG received her B.Eng. and M.Eng. degrees from the Southwest University of Science and Technology (SWUST), China, in 2003 and 2006, respectively, and the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China (UESTC) in 2012. From August 2011 to January 2012, she was an international visitor at the Institute for Infocomm Research (I2R), Singapore. She is currently an associate professor in School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China. Her research interests include wireless communication and optimization, wireless network virtualization, mobile edge computing and IoT.

DEMING MAO is currently pursuing the Ph.D. degree with the College of Cyberspace Security, Northwestern Polytechnical University, Xi'an, Shanxi, China. He is a Senior Engineer with China Electronic Technology Cyber Security Company, Ltd., Chengdu, Sichuan, China. His research interest includes cyberspace security.

LEI ZHANG received the Ph.D. degree from The University of Sheffield, U.K. He is currently a Lecturer at the University of Glasgow, U.K. His research interests include wireless communications and networks, blockchain, radio access network slicing (RAN slicing), the Internet of Things (IoT), multiantenna signal processing, and MIMO systems. He has 19 U.S./U.K./EU/Chinese granted/filed patents on wireless communications and published two books and more than 100 peerreviewed papers. He received the IEEE Communication Society TAOS Best Paper Award in 2019. He is the Technical Program Chair of 5th UK-China Emerging Technologies (UCET) in 2020. He was the Publication and Registration Chair of the IEEE Sensor Array and Multichannel (SAM) in 2018, the Publicity Chair of 4th UK-China Emerging Technologies (UCET) in 2019, and the Co-Chair of Cyber-C Blockchain Workshop in 2019. He is an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, the IEEE WIRELESS COMMUNICATIONS LETTERS, and IEEE ACCESS.