SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2018, Vol. 61 102304:1–102304:11
https://doi.org/10.1007/s11432-017-9296-6

# Enhanced frameless slotted ALOHA protocol with Markov chains analysis

Dai JIA[1], Zesong FEI[1*], Ming XIAO[2], Congzhe CAO[3] & Jingming KUANG[1]

[1]*School of Information and Electronics, Beijing Institute of Technology, Beijing* 100081*, China;*
[2]*School of Electrical Engineering, Royal Institute of Technology, Stockholm* SE-100 44*, Sweden;*
[3]*Department of Electrical and Computer Engineering, University of Alberta, Edmonton* T6G2R3*, Canada*

Received 4 June 2017/Revised 23 September 2017/Accepted 22 November 2017/Published online 30 May 2018

**Abstract** In this paper, we propose a novel approach to enhance the performance of frameless slotted ALOHA (SA) protocol. We employ signature codes to help the receiver identify the packets contained in collisions, and use successive interference cancellation (SIC) for packet recovery. We model the proposed scheme as a two-state Markov model represented by a uni-partite graph. We evaluate the throughput, expected delay and average memory size of the proposed scheme, and optimize the proposed scheme to maximize the throughput. We show that the theoretical analysis matches well with simulation results. The throughput and expected delay of the proposed protocol outperform the conventional slotted ALOHA protocol significantly.

**Keywords** slotted ALOHA, SIC, Markov chain, uni-partite graph, throughput, delay

**Citation** Jia D, Fei Z S, Xiao M, et al. Enhanced frameless slotted ALOHA protocol with Markov chains analysis. Sci China Inf Sci, 2018, 61(10): 102304, https://doi.org/10.1007/s11432-017-9296-6

## 1 Introduction

In machine-to-machine (M2M) communications, a random number of devices transmit data over a shared channel in the same time slots and frequency bands [1, 2]. Since collisions may occur due to independent transmission of each device, it is critical to study efficient multiple access techniques. Slotted ALOHA (SA) protocol [3, 4] is widely used for multiple access networks [5–7]. In this protocol, transmission of users is divided into slots and starts only at the beginning of each slot, where users can simultaneously transmit their packets, resulting in collisions. Therefore, it is essential to manage the collisions in order to increase the system throughput [8].

There has been a great interest in improving the throughput of the SA protocol in the past decades. In [9], the compressive sensing method is employed to reduce the probability of collisions. Forward error correction (FEC) schemes are employed to recover packets from collisions in [10, 11]. In [3, 12, 13], the capture effect benefited from different power levels of received signals is exploited, such that multiple packets can be recovered from collisions with multiuser detection (MUD) methods, which separates the collided packets and recovers multiple packets within the same slot, at the cost of higher complexity. Meanwhile, transmission diversity combined with successive interference cancellation (SIC) is proposed as another collision resolution approach in [14]. When a collided packet is received, it is stored and the receiver waits for clean copies of packets involved in this collided packet, which are used to cancel

* Corresponding author (email: feizesong@bit.edu.cn)

 

interference in the collided packet. This scheme is further exploited in irregular repetition slotted ALOHA (IRSA) [15], which transmits a variety number of replicas of the same packet in a frame. The optimization of IRSA scheme is proposed in [16].

In the schemes with the SIC algorithm, each user employs a repetition code to generate $r$ replicas of its own packet, and then transmits them in $r$ time slots, which are chosen uniformly at random within a fixed length frame. Each data packet is equipped with pointers to other $r-1$ replicas. When a clean copy of any packet is received, the receiver can obtain the position of all its replicas in this frame. Then the receiver cancels this packet from collided packets. However, those schemes have two drawbacks. One is that the SA protocol has to set the frame length beforehand and the frame synchronization method should be employed for all the users [17]. As a result, users have to transmit a packet many times during one frame even if it has been recovered at the receiver. The other drawback is that the pointer increases the overhead of the system, especially with increasing frame length and number of replicas [18].

In this paper, we propose a novel approach to improve the throughput of SA protocol by employing signature codes [19] to replace the pointer as the header of the transmitted packet. As a result, the receiver can process the received packet with no more than $K$ colliding users and identify the packets contained in this collided packet once it is received. It enables the SIC algorithm to be employed in the frameless SA protocol. We then analyze the minimum length of signature codes which is determined by the maximum detectable number of users $K$. We use a uni-partite graph to represent the states of the received packets, i.e., a free state and a backlogged state. Then, we model the system as a two-state Markov model [20, 21] by a stochastic game [22] where the number of packets in backlogged state $N$ is taken as the system state. We derive the stationary distribution and give the theoretical analysis of the throughput, expected delay and average memory size. The explicit expressions are given as a function of the first transmission and retransmission probability. The retransmission probability is optimized for a given first transmission probability to maximize the throughput. Finally, numerical results show good agreement with the theoretical analysis and that the proposed protocol significant outperforms the conventional SA protocol.

Our contributions are summarized as follows.

- We employ signature codes as headers of data packets, which enable the SIC algorithm to be employed in the frameless SA protocol.
- Based on the proposed protocol, we use a uni-partite graph to represent the full states of the received packets.
- We use a Markov model to evaluate the throughput and expected delay of the proposed protocol, and validate them with simulation results.

The rest of this paper is organized as follows. We first give a brief introduction to the SA system model in Section 2. Section 3 discusses using the uni-partite graph to represent the proposed protocol. The Markov chain analysis is given in Section 4. Section 5 presents the performance of the proposed protocol. Finally, some concluding remarks are provided in Section 6.

## 2 System model

We consider an SA network where $M$ active users attempt to transmit their packets to the same receiver, as illustrated in Figure 1. All users are synchronized and can start transmission only at the beginning of each time slot with duration $T_{\text{slot}}$. If a user successfully transmits its packet, it waits for a new generated packet in the following time slot. We denote the first transmission probability $p_f$ as the probability that a new packet generated by a user. During transmission, if a user detects a collision, it retransmits the previous packet in each subsequent time slot with a retransmission probability $p_r$ until the transmission succeeds. We call $p_r$ retransmission probability. We assume the generation of user packets is independent and if the current transmission of a user fails, new packets to that user are blocked and lost.

The SA protocol described above can be modeled as a two state system, where the states indicate the outcome of the previously attempted transmission according to the feedback from the receiver. A user

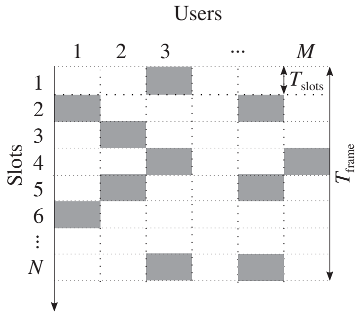


**Figure 1** Slotted ALOHA system.

is in the free state if its previous transmission succeeds; otherwise the user is in the backlogged state. As a result, in the free state, a user transmits a new packet in the next slot with probability $p_f$. In the backlogged state, it transmits in the next slot with probability $p_r$.

Let $u_{m,i}$ be the $i$-th packet from user $m$, $1 \leqslant m \leqslant M$. Channel coefficient is constant in each slot, but changes among different slots due to fading. The receiver obtains the superposition signal

$$y_j = \sum_{m=1}^{M} c_{m,j} h_{m,j} u_{m,i} + n_j, \tag{1}$$

where $h_{m,j}$ represents the fading coefficient between the $m$-th user and the receiver in time slot $j$. $n_j$ is the additive white Gaussian noise (AWGN), with zero mean and variance $\sigma_n^2$. $c_{m,j} = 1$ if $u_{m,i}$ is transmitted in time slot $j$, otherwise $c_{m,j} = 0$. For simplifying illustration, we ignore the subscript $i$ in $u_{m,i}$. We assume proper channel coding and ideal channel estimation at the receiver. This assumption is to guarantee that we can recover the transmitted packet if there is no collision, for each time slot. If multiple packets collide in one time slot, the receiver is able to identify the packets contained in the collisions by employing signature codes.

**Signature codes.** Note that signature codes introduce the cost to the system, we calculate the minimum length of the signature codewords.

Each user $m$ is provided with a unique signature codeword $s_m$. The length of signature code $L$, which guarantees that a sum of up to $K$-out-of-$M$ signature codewords is uniquely decodable [19], is given by

$$L = \lceil \log_2(M^K - 1) \rceil + 1 \approx \lceil K \log_2 M \rceil. \tag{2}$$

Obviously, the length of the signature code is proportional to $K$. The larger $K$ leads to larger header, resulting in larger overhead. Note that most M2M applications transmit and receive small amount of information data [23, 24]. If $K$ increases, it leads to an unreasonable ratio between information data and signature codes. To keep the header short, we only employ the received packets with no more than 2 colliding users, and the packets with larger $K$ will be discarded. The length of the signature code is

$$L = \lceil \log_2(M^2 - 1) \rceil + 1. \tag{3}$$

Clearly, this work can be extended to the case with larger $K$ at a cost of overhead.

## 3 The proposed receiver for slotted ALOHA

In this section, we propose a novel approach to enhance the performance of frameless SA protocol. The proposed protocol employs the received packets with no more than 2 colliding users to recover the user packets. By employing signature codes, the receiver can process the received packets with no more than 2 colliding users and identify the packets contained in the collided packet once it is received. This protocol only requires slot synchronization and it is suitable for the frameless system.

We demonstrate four conditions based on the feedback from receivers, as shown in Figure 2.

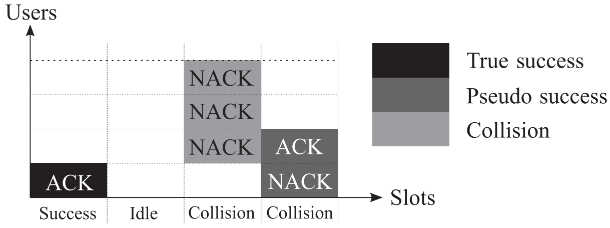


**Figure 2** Different kinds of feedback.

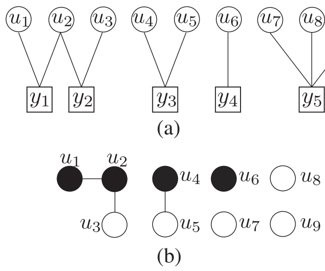


**Figure 3** (a) Bi-partite graph; (b) uni-partite graph.

(1) Idle. No user is active in a time slot. No feedback is given by the receiver.

(2) True success. Only 1 user is active in a time slot. The receiver transmits acknowledgement (ACK) to this user.

(3) Collision. More than 2 users are active in a time slot. The receiver transmits negative acknowledgment (NACK) to these users.

(4) Pseudo success. 2 users are active in a time slot. The receiver transmits the feedback messages as follows:

(a) If both users are in the free state/backlogged state, we choose a user at random to receive ACK and the other receive NACK;

(b) If one user is in the free state and the other one is in the backlogged state, the receiver transmits ACK to the backlogged user and NACK to the unbacklogged user.

If the receiver is in the pseudo success condition, the received packet is unrecovered when it is received. The receiver stores it and waits for a clean copy of packets involved in this collided packet. Then we can recover the stored packet by using SIC algorithm.

To analyze the full states of user packets, we employ a uni-partite graph to describe the proposed protocol as follows. The SA protocols can be represented by a bi-partite graph, as shown in Figure 3(a), where user nodes $\boldsymbol{u} = [u_1, u_2, \ldots, u_m, \ldots]$ represent the user packets; slot nodes $\boldsymbol{y} = [y_1, y_2, \ldots, y_j, \ldots]$ represent the received packets at the receiver. An edge between a user node $u_m$ and a slot node $y_j$ indicates $u_m$ is one of the summands of the received packet $y_j$. Since the bi-partite graph cannot directly represent the full users states, we employ a uni-partite graph to represent the full user states, as shown in Figure 3(b). A uni-partite graph represents a collection of received packets which are the sum of $K$ users packets, $K \leqslant 2$. In the uni-partite graph, a node is colored black (or white) if it is in the free (or backlogged) state. Two nodes are connected with an edge if there is a received packet with 2 colliding users. The set of connected packets is denoted as a component, e.g., $[u_1, u_2, u_3]$ or $[u_4, u_5]$ in Figure 3 (b). Each component contains only one white node. Note that user node turns into black once a clean copy of corresponding user packet is received. This user can transmit a new packet to the receiver. If there is a received packet with more than 2 colliding users, all the user nodes involved in this received packet turn to white.

The proposed protocol follows a simple observation that if a received packet adds an edge between two white nodes, the connected components of the two nodes (if not already in the same component) are merged to a single component. More importantly, if a white node turns to black, all the nodes in the corresponding component can be recovered with the SIC algorithm. For example, in Figure 3 (b), if the receiver gets a clean copy of $u_3$, we can recover $u_1$, $u_2$, $u_3$.

Based on the proposed SA protocol above, a Markov model is employed to analyze and optimize $p_f$ and $p_r$ in order to maximize the throughput of the slotted ALOHA protocol.

## 4 Performance analysis with Markov chains

### 4.1 Problem formulation

In this subsection, we give theoretical analysis of the proposed SA protocol in terms of the throughput, delay and required memory size. We denote $N$ as the number of backlogged users at the beginning of a slot, where $0 \leqslant N \leqslant M$. We assume that $p_f$ is different from $p_r$. Let $P_r(i,N)$ be the probability that $i$ out of the $N$ backlogged packets are retransmitted in a given slot which can be expressed as

$$P_r(i,N) = \binom{N}{i}(1-p_r)^{N-i}p_r^i. \tag{4}$$

Let $P_f(i,N)$ be the probability that $i$ unbacklogged users (users in free state) transmit their packets in a given slot. Then

$$P_f(i,N) = \binom{M-N}{i}(1-p_f)^{M-N-i}p_f^i. \tag{5}$$

As a result, the probabilities of four immediate conditions with $N$ backlogged users are respectively given by

$$P_I(N) = P_f(0,N)P_r(0,N), \tag{6}$$
$$P_{\mathrm{TS}}(N) = P_f(1,N)P_r(0,N) + P_f(0,N)P_r(1,N), \tag{7}$$
$$P_{\mathrm{PS}}(N) = P_f(2,N)P_r(0,N) + P_f(0,N)P_r(2,N) + P_f(1,N)P_r(1,N), \tag{8}$$
$$P_C(N) = 1 - P_I - P_{\mathrm{TS}} - P_{\mathrm{PS}}. \tag{9}$$

In what follows, we use a Markov model to analyze the proposed protocol. We denote the number of backlogged users $N$ as the state. The transition probability of the Markov chain is given by

$$P_{N,N+i} = \begin{cases} P_f(i,N), & \text{for } i \geqslant 3, \\ P_f(2,N)[1-P_r(0,N)], & \text{for } i = 2, \\ P_f(1,N)[1-P_r(0,N)-P_r(1,N)] + P_f(2,N)P_r(0,N), & \text{for } i = 1, \\ P_f(0,N)[1-P_r(1,N)-P_r(2,N)] + P_f(1,N)[P_r(0,N)+P_r(1,N)], & \text{for } i = 0, \\ P_f(0,N)P_r(1,N) + P_f(0,N)P_r(2,N), & \text{for } i = -1, \end{cases}$$

where $P_{N,N+i}$ denotes the probability that the number of backlogged users change from $N$ to $N+i$ after receiving a new packet. Since the state space is finite and all the states communicate among themselves, the Markov chain is ergodic. We denote $\pi(p_f,p_r)$ as the steady state probabilities, where its $N$-th entry, $\pi_N(p_f,p_r)$, denotes the probability that there are $N$ backlogged users in a given time slot. This steady state distribution can be obtained by solving the following problem:

$$\begin{cases} \pi_N(p_f,p_r) \geqslant 0, \qquad N \in [0,M], \\ \displaystyle\sum_{N=0}^{M} \pi_N(p_f,p_r) = 1, \\ \pi_N(p_f,p_r) = \displaystyle\sum_j \pi_j(p_f,p_r)P_{j,N}. \end{cases}$$

### 4.2 Throughput

In this subsection, we attempt to maximize the throughput of the proposed scheme which employs the received packets with no more than 2 colliding users. The throughput $T$ is defined as the number of successfully recovered packets in each slot, which is a function of first transmission probability $p_f$ and retransmission probability $p_r$.

**Proposition 1.** The throughput is given by

$$T = \sum_{N=0}^{M} \pi_N(p_f, p_r)(P_{\rm TS}(N) + P_{\rm PS}(N)) \tag{10}$$

$$= p_f \left( M - \sum_{N=0}^{M} \pi_N(p_f, p_r) N \right), \tag{11}$$

where $N$ denotes the number of backlogged packets and $M$ denotes the number of active users.

*Proof.* Note that the expected number of ACKs equals the expected number of successful transmitted packets at the steady state. Since true success and pseudo success conditions feed back ACK, the probability that an ACK is fed back with $N$ backlogged users can be expressed as

$$\begin{aligned} P_{\rm ACK}(N) =& P_{\rm TS}(N) + P_{\rm PS}(N) \\ =& P_f(1,N)P_r(0,N) + P_f(0,N)P_r(1,N) + P_f(2,N)P_r(0,N) \\ &+ P_f(1,N)P_r(1,N) + P_f(0,N)P_r(2,N). \end{aligned} \tag{12}$$

The throughput is the sum of $P_{\rm ACK}(N)$ with all values of $N$, which is given by

$$T = \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\rm ACK}(N) = \sum_{N=0}^{M} \pi_N(p_f, p_r)(P_{\rm TS}(N) + P_{\rm PS}(N)). \tag{13}$$

Then the proof of (10) is completed.

Moreover, the expected number of packet arrivals equals the expected number of successful transmitted packets at the steady state. The throughput can be written as follows:

$$T = p_f(M - R), \tag{14}$$

where $R$ denotes the average number of backlogged users which is given by

$$R = \sum_{N=0}^{M} \pi_N(p_f, p_r) N. \tag{15}$$

Then the proof of (11) is completed.

Therefore, the solution to maximize the throughput is given by

$$\max_{p_f, p_r} T \quad \text{s.t.} \begin{cases} \pi_N(p_f, p_r) \geqslant 0, \qquad N \in [0, M], \\ \displaystyle\sum_{N=0}^{M} \pi_N(p_f, p_r) = 1, \\ \pi_i(p_f, p_r) = \displaystyle\sum_{j} \pi_j(p_f, p_r) P_{j,i}. \end{cases}$$

### 4.3 Expected successful transmission probability

Note that failed transmission of the current packet leads to the blocking of new arrived packets. As a result, we have to reduce the percentage of blocked packets.

**Proposition 2.** The expected probability of successful transmission is the ratio of throughput to the expected number of arrival packets, which is given by

$$P = \frac{M - R}{M}. \tag{16}$$

*Proof.* For the definition of $P$, we have

$$P = \frac{T}{p_f M} = \frac{p_f(M - R)}{p_f M} = \frac{M - R}{M}. \tag{17}$$

Thus, the maximization of $P$ is equivalent to the minimization of $R$. Therefore, the solution to maximize the probability of successful transmission is given by

$$\min_{p_f, p_r} R \quad \text{s.t.} \begin{cases} \pi_N(p_f, p_r) \geqslant 0, \qquad N \in [0, M], \\ \displaystyle\sum_{N=0}^{M} \pi_N(p_f, p_r) = 1, \\ \pi_i(p_f, p_r) = \displaystyle\sum_j \pi_j(p_f, p_r) P_{j,i}. \end{cases}$$

Note that the optimized pair of $[p_f, p_r]$ is different from the solution to maximize the throughput. However, for a given $p_f$, the maximization of $T$ is equivalent to the minimization of $R$, as shown in (14). That is, the above two solutions have the same optimized $p_r$.

### 4.4 Expected delay of transmitted packets

We define the expected delay of transmitted packets as the number of time slots from a packet transmitted for the first time to this packet is recovered by the receiver successfully. By employing Little's formula [25], the expected delay of transmitted packets is given by

$$D = 1 + R/T. \tag{18}$$

### 4.5 Average memory size

Since the proposed algorithm stores the received packets with no more than 2 colliding users and waits for a clean copy of packets involved in these packets, the receiver requires a memory space to store these packets. Therefore, we evaluate the average memory size $S$ required for the proposed scheme.

**Proposition 3.** The average memory size $S$ is given by

$$S = \frac{\sum_{N=0}^{M} \pi_N(p_f, p_r) N \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{st}}}{\sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{re}}}, \tag{19}$$

where $P_{\text{st}}$ denotes the probability of a received packet stored in the memory and $P_{\text{re}}$ denotes the probability of a component of received packets released from the memory at the steady state.

*Proof.* Note that the expected number of packets stored in the memory equals the expected number of packets released from the memory at the steady state.

Since $P_{\text{st}}$ is denoted as the probability of a received packet stored in the memory, namely the probability of a received packet with 2 colliding users, we have

$$P_{\text{st}} = P_f(2, N) P_r(0, N) + P_f(1, N) P_r(1, N) + P_f(0, N) P_r(2, N). \tag{20}$$

The expected number of packets stored in the memory for each slot is given by

$$S_{\text{in}} = \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{st}}. \tag{21}$$

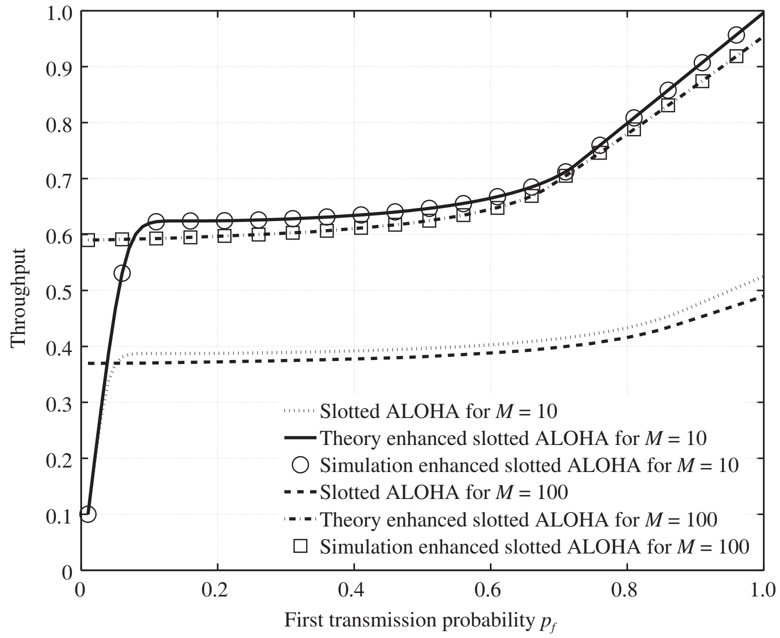


**Figure 4** Throughput for different first transmission probability $p_f$.

As explained in Section 3, if a user node turns to free state, all the packets in the corresponding component can be recovered with the SIC algorithm. The expected number of packets released from the memory for each slot is given by

$$S_{\text{out}} = \frac{S \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{re}}}{R}, \tag{22}$$

where $P_{\text{re}} = P_f(0, N)P_r(1, N)$.

Note that $S_{\text{in}} = S_{\text{out}}$. As a result, the expected number of packets stored in the memory is given by

$$S = \frac{R \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{st}}}{\sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{re}}} = \frac{\sum_{N=0}^{M} \pi_N(p_f, p_r) N \sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{st}}}{\sum_{N=0}^{M} \pi_N(p_f, p_r) P_{\text{re}}}. \tag{23}$$

## 5 Numerical results

In this section, we present the numerical results of the proposed SA protocol. To demonstrate the theoretical analysis of the proposed protocol, we set the number of active users $M = 10$ and 100 and compare the performance with conventional SA protocol. First, we show the throughput performance as a function of $p_f$. Then, $p_r$ is optimized for given $p_f$. After that, we compare the expected delay of transmitted packets and investigate the required average memory size.

We optimize $p_r$ given each $p_f$. $p_f$ and $p_r$ ranges from 0.01 to 1 and 0.001 to 1, respectively. Figure 4 provides the throughput as a function of $p_f$. As suggested in [26], we set the length of data packet in each time slot to $L_p = 800$ bits. By taking the length of signature codes into account, the actual throughput of the proposed protocol is given by

$$T_{\text{act}} = \frac{L_p - L}{L_p} T, \tag{24}$$

where $L$ is the length of signature codes. In Figure 4, it can be seen that simulation results match well with the analysis. We can see that the throughput of the proposed protocol outperforms the conventional SA protocol, significantly. The optimized throughput is improved by 89.7% with $p_f = 1$. This probability means that all users always have packets to send. Moreover, we see that in heavy traffic, e.g., $p_f \geqslant 0.1$, the throughput becomes lower with the increasing $M$.

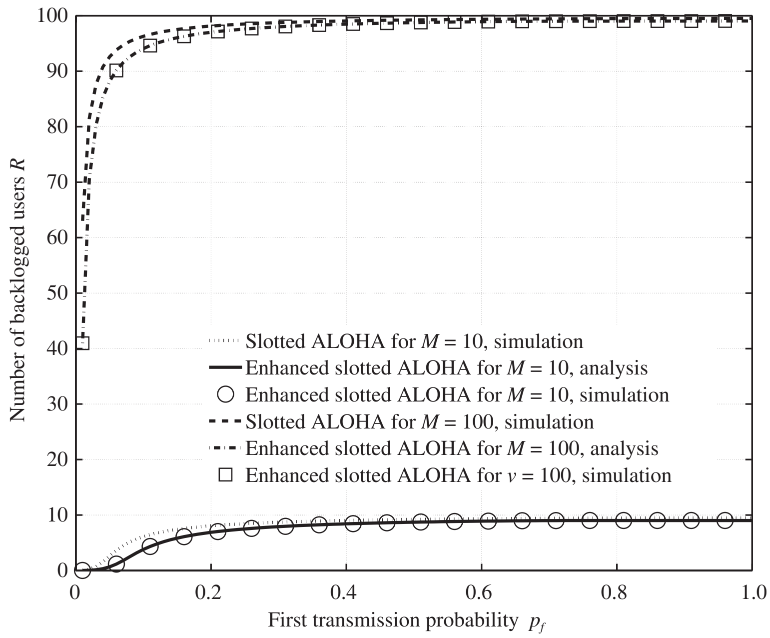


**Figure 5** Number of backlogged users for different first transmission probability $p_f$.

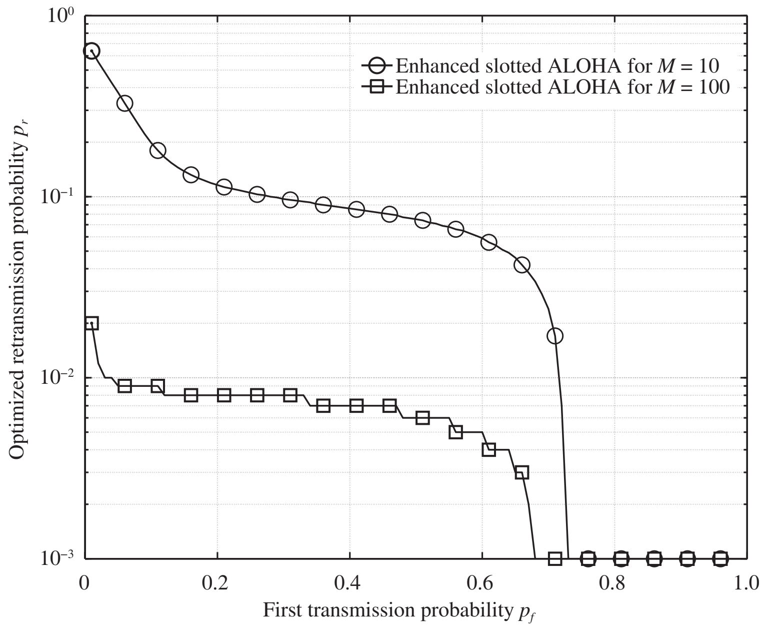


**Figure 6** Optimized retransmission probability $p_r$ for different $p_f$.

Figure 5 provides the number of backlogged users as a function of $p_f$. The minimum number of backlogged users for $M = 10$ and $M = 100$ are obtained at $p_f = 0.01$. It is demonstrated that higher probability of successful transmission can be obtained with smaller $M$. Thus, the solution for minimizing $R$ is opposite to maximizing $T$. In practice, we have to make a trade off between these two solutions.

The optimal $p_r$ is shown in Figure 6. We can see that the optimal $p_r$ decreases with increasing $p_f$ or $M$. In other words, as the system becomes more congested (increasing $p_f$ or $M$), $p_r$ decreases to counter expected collisions. We can see that $p_r \to 0$ when $p_f \to 1$. As a result, if a user finds all other users are backlogged, it can transmit its packets for very long time without collisions. Since the user $p_f$ is close to 1, the channel resource is not wasted during such periods.

The expected delay of transmitted packet is shown in Figure 7. The simulation results show good

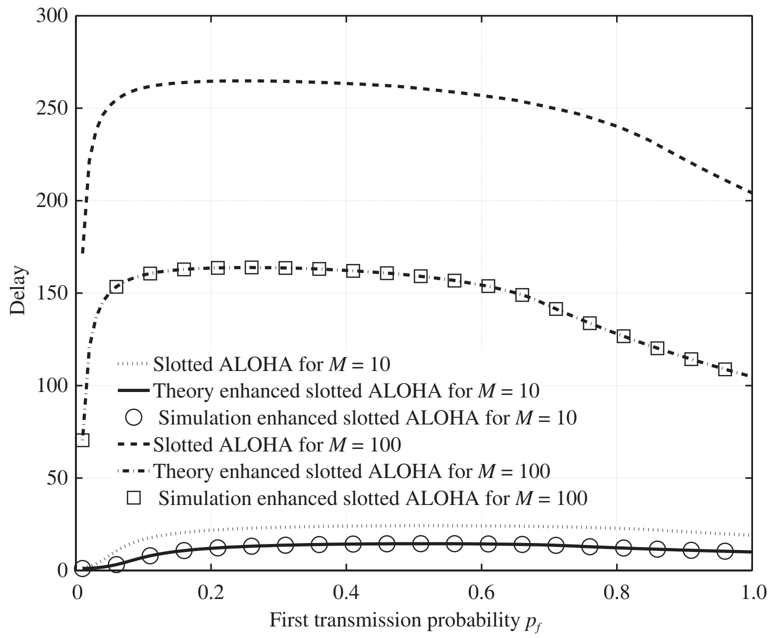


**Figure 7** Expected delay of transmitted packets for different $p_f$.

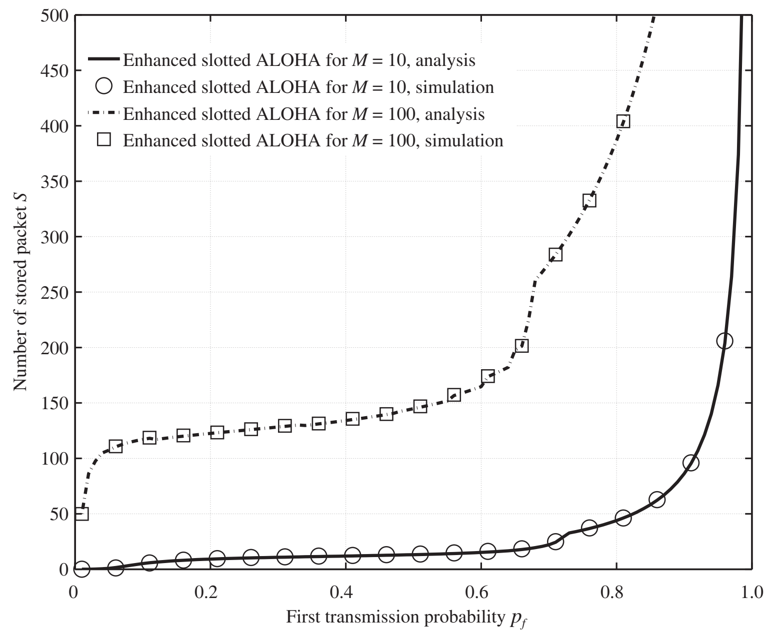


**Figure 8** Average memory size for different $p_f$.

agreement with the theoretical analysis of the proposed protocol. We can see that the enhanced SA protocol reduces the delay of transmitted packet significantly, compared to the conventional SA protocol. The expected delay of transmitted packets decreases when $p_f$ is beyond a certain value, which depends on the number of active users, e.g., for $p_f = 0.4$, $M = 100$ and $p_f = 0.6$, $M = 10$.

Figure 8 shows the average memory size of the proposed protocol. Simulation results show that the average memory size increases with increasing $p$ or $M$. The memory size keeps at a low level below a certain value of $p_f$ which depends on the number of active users, e.g., $p_f = 0.65$, $M = 100$ and $p_f = 0.8$, $M = 10$. After that value, the memory size increases rapidly. This analysis helps set proper storage space for the receiver according to different $p_f$ or $M$.

## 6 Conclusion

We proposed an enhanced frameless slotted ALOHA protocol. By employing signature codes, the proposed algorithm could exploit the received packets with no more than $K$ colliding users. A uni-partite graph was used to represent the full states of users. Based on that, we constructed a Markov model and analyzed its stationary distribution to evaluate the performance of the proposed protocol. The first transmission and retransmission probability were optimized to maximize the throughput or to minimize the average number of backlogged users. Moreover, the expected delay and average memory size were investigated. Simulation results match well with analysis and the proposed protocol outperformed the conventional slotted ALOHA protocol.

**Acknowledgements** This work was supported in part by Beijing Major Science and Technology Projects (Grant No. D171100006317001), Beijing Natural Science Foundation (Grant No. 4152047), and Programme of Introducing Talents of Discipline to Universities (111 Project) (Grant No. B14010).

### References

1 IEEE 802.16p-10/0005. Machine-to-Machine (M2M) Communication Study Report. 2010
2 Yan S, Peng M G, Abana M A, et al. An evolutionary game for user access mode selection in fog radio access networks. IEEE Access, 2017, 5: 2200–2210
3 Roberts L G. ALOHA packet system with and without slots and capture. ACM SIGCOMM Comput Commun Rev, 1975, 5: 28–42
4 Ghez S, Verdu S, Schwartz S C. Stability properties of slotted ALOHA with multipacket reception capability. IEEE Trans Autom Control, 1988, 33: 640–649
5 Zheng J, Li J D, Liu Q, et al. Performance analysis of three multi-radio access control policies in heterogeneous wireless networks. Sci China Inf Sci, 2013, 56: 122305
6 Xu D, Li Q. Price-based time and energy allocation in cognitive radio multiple access networks with energy harvesting. Sci China Inf Sci, 2017, 60: 108302
7 Wei N, Zhang Z P. Competitive access in multi-RAT systems with regulated interference constraints. Sci China Inf Sci, 2017, 60: 022306
8 Namislo C. Analysis of mobile radio slotted ALOHA networks. IEEE J Sel Areas Commun, 1984, 2: 583–588
9 Wu L T, Sun P, Xiao M, et al. Sparse signal ALOHA: a compressive sensing-based method for uncoordinated multiple access. IEEE Commun Lett, 2017, 21: 1301–1304
10 Herrero O R, Gaudenzi R D. Generalized analytical framework for the performance assessment of slotted random access protocols. IEEE Trans Wirel Commun, 2014, 13: 809–821
11 Zhang Z H, Xu C B, Ping L. Coded random access with distributed power control and multiple-packet reception. IEEE Wirel Commun Lett, 2015, 4: 117–120
12 Zhang Y, Wang H M, Zheng T X, et al. Energy-efficient transmission design in non-orthogonal multiple access. IEEE Trans Veh Technol, 2017, 66: 2852–2857
13 Han W J, Zhang Y, Wang X J, et al. Orthogonal power division multiple access: a green communication perspective. IEEE J Sel Areas Commun, 2016, 34: 3828–3842
14 Casini E, De Gaudenzi R, Herrero O R. Contention resolution diversity slotted ALOHA (CRDSA): an enhanced random access schemefor satellite access packet networks. IEEE Trans Wirel Commun, 2007, 6: 1408–1419
15 Paolini E, Liva G, Chiani M. Graph-based random access for the collision channel without feedback: capacity bound. In: Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), Houston, 2011
16 Sun Z, Xie Y X, Yuan J H, et al. Coded slotted ALOHA schemes for erasure channels. In: Proceedings of IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016
17 Stefanovic C, Popovski P, Vukobratovic D. Frameless ALOHA protocol for wireless networks. IEEE Commun Lett, 2012, 16: 2087–2090
18 Jia D, Fei Z S, Lin H, et al. Distributed decoding for coded slotted ALOHA. IEEE Commun Lett, 2017, 21: 1715–1718
19 Goseling J, Stefanovic C, Popovski P. Sign-compute-resolve for random access. In: Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, 2014. 675–682
20 Nelson R. Probability, Stochastic Process, and Queueing Theory. Berlin: Springer, 2000
21 Vajargah B F, Gharehdaghi M. Ergodicity of fuzzy Markov chains based on simulation using Halton sequences. J Math Comput Sci, 2014, 11: 159–165
22 Altman E, Azouzi R E, Jiménez T. Slotted ALOHA as a game with partial information. Comput Netw, 2004, 45: 701–713
23 Gotsis A G, Lioumpas A S, Alexiou A. M2M scheduling over LTE: challenges and new perspectives. IEEE Veh Technol Mag, 2012, 7: 34–39
24 Wang K, Du M, Sun Y F, et al. Attack detection and distributed forensics in machine-to-machine networks. IEEE Netw, 2016, 30: 49–55
25 Little J D C. A proof for the queuing formula: $L = \lambda W$. Oper Res, 1961, 9: 383–387
26 ETSI GS LTN 001. Low Throughput Networks (LTN); Use Cases for Low Throughput Networks. V1.1.1 (2014-09)