Ad Hoc Networks 42 (2016) 34–46

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Performance analysis of Block ACK-Based Slotted ALOHA for wireless networks with long propagation delay☆

Hoki Baek[a], Jaesung Lim[b,*]

[a] Department of Military Digital Convergence, Ajou University, Suwon, South Korea
[b] Graduate School of Computer Engineering, Ajou University, Suwon, South Korea

ARTICLE INFO

Article history:
Received 24 August 2015
Revised 11 December 2015
Accepted 22 January 2016
Available online 2 February 2016

Keywords:
Framed structure
Performance analysis
Retransmission

ABSTRACT

Recently, many variants of Slotted ALOHA (S-ALOHA) have been proposed to solve a problem of performance degradation in wireless networks with long propagation delay. However, they do not consider the effect of retransmission, which also largely degrades performance, and do not provide any analytical model that considers the effect. In this paper, we design a variant of S-ALOHA to support retransmission and derive analytical models that do consider its effect. The designed scheme has a framed structure that starts with a coordinators beacon. The beacon consists of a coordinators timestamp and Block ACKnowledgment (B-ACK). Using the timestamp, a node estimates propagation delay to the coordinator (*PDC*) in order to reduce guard time while transmitting a packet. Moreover, B-ACK is used to report the results of all transmissions attempted in the previous frame at once. As a result, the designed scheme can largely reduce the number of feedbacks and waste of guard time. Even if there is no analytical model that considers the long propagation delay and retransmission simultaneously, we choose the existing analytical models that consider a framed structure and B-ACK as reference models. However, they are not fully mathematical and partially use simulation results because of high computational complexity. Moreover, these models only analyze stability and throughput as performance metrics. On the other hand, our analytical models are fully mathematical models and can analyze all metrics, such as stability, throughput, and packet delay. We expect our analytical models to be a foundation for deriving fully mathematical models for variants of S-ALOHA using a framed structure and B-ACK.

1. Introduction

Slotted ALOHA (S-ALOHA) is one of the most widely used random access protocols originally designed to enhance ALOHA throughput. While all the nodes in an ALOHA network transmit a packet immediately after it is generated, the nodes in an S-ALOHA network first wait until the start of the upcoming slot to transmit the packet. Thus, S-ALOHA can provide higher throughput than ALOHA because of reduced packet collision probability.

However, this is not valid when the propagation delay of a network is not negligible. In such network, there are collisions caused not only by time uncertainty, but also by long propagation delay. This is called the space-time uncertainty [1,2]. To prevent collisions caused by the long propagation delay, a guard time is required. In [3], simulation result shows that ALOHA and S-ALOHA without guard time provides almost equal throughput because S-ALOHA without guard time suffers from collisions caused by not only

☆ This work was supported by National Research Foundation of Korea (NRF) grant funded by Korean Government (MSIP) (no. 2013R1A2A10101643).

* Corresponding author. Tel.: +82 312192545; fax: +82 312191842.
E-mail addresses: neloyou@ajou.ac.kr (H. Baek), jaslim@ajou.ac.kr (J. Lim).

http://dx.doi.org/10.1016/j.adhoc.2016.01.004
1570-8705/

time uncertainty but also long propagation delay. After all, the throughput of S-ALOHA can be degraded. In [2], S-ALOHA with guard time is considered to resolve the space-time uncertainty. However, the results show that the guard time can be a waste and degrades throughput of S-ALOHA. In [1,4], analytical model for normalized throughput of S-ALOHA is given. Under the assumption that a packet arrival from infinite nodes follows Poisson distribution and retransmission is not considered, normalized throughput of S-ALOHA is given by $Ge^{G(1+\alpha)}$, where G is the offered load and α is the ratio of a guard time length to a packet transmission delay.

We define a *long propagation network* as the network with non-negligible α and different propagation delays between a coordinator and any nodes. In the long propagation network, normalized throughput of S-ALOHA is degraded and even lower than that of ALOHA when α is greater than one [1].

Recently, many variants of S-ALOHA have been proposed to solve the problem of throughput degradation in long propagation networks [1–3,5–8]. In [5], Improved Synchronized Arrival Slotted ALOHA (ISA-ALOHA) provides a time alignment mechanism where a node adjusts the start time of transmission in order to allow a packet to arrive at the start time of a slot at the coordinator. The time alignment mechanism can prevent collisions caused by different propagation delays. Thus, ISA-ALOHA can provide higher throughput than S-ALOHA. In [6], the modified Receiver Synchronized S-ALOHA (mRSS-ALOHA-uw) also performs time alignment mechanism to reduce the guard time. Moreover, it also considers the imperfect propagation delay information owing to the irregular underwater signal propagation speed. Thus, various guard time lengths are used. However, both ISA-ALOHA and mRSS-ALOHA-uw require the assumption that all nodes know their *propagation delay to coordinator* (*PDC*), even if the PDC is not accurate. In [1], Beacon-Based Slotted ALOHA (BS-ALOHA) does not require such assumption. Instead, it provides a framed structure to estimate *PDC*. At the start of every frame, the coordinator broadcasts a beacon with its timestamp. After receiving the beacon, all nodes estimate *PDC* periodically. Then all nodes can perform time alignment with the *PDC*. After all, BS-ALOHA can improve normalized throughput without the assumption of both ISA-ALOHA and mRSS-ALOHA-uw. More variants of S-ALOHA are surveyed and categorized in [7,8,16].

However, existing schemes, such as BS-ALOHA and ISA-ALOHA, have limitations. First, these schemes do not consider retransmission, although it can largely degrade performance. If these schemes support retransmission, sending a feedback, ACKnowledgment (ACK), requires non-negligible guard time again. However, this guard time cannot be reduced by the time alignment mechanism proposed in [1,5]. Thus, we need to provide a method for minimizing the effect of sending feedback when we consider a retransmission in S-ALOHA. Second, there is no adequate analytical model that considers the effect of retransmission in long propagation networks. Moreover, a new variant of S-ALOHA may have a framed structure as BS-ALOHA. Then, the analytical model should additionally consider not only the framed structure, but also the effect of retransmission in long propagation networks. However, there is no such analytical model yet.

More specifically, there are several analytical models for S-ALOHA that consider retransmission. In [9,10], S-ALOHA supports retransmission with an immediate ACK. In a slot, a sender transmits a packet and the receiver immediately sends an ACK if it receives the packet successfully. These models were constructed using a Markov chain whose state is defined as the number of backlogged nodes. The state can be changed individually by slot. In [11,12], S-ALOHA has a framed structure and supports retransmission with a delayed ACK. In [11], if a receiver obtains a packet successfully, it sends an ACK after several slots. Thus, ACK corresponds to a packet. On the other hand, in [12], ACK is different from that of [11] because this reports the results of all transmissions attempted by all nodes in the previous frame at once. Thus, this ACK is called Block ACK (B-ACK). At the beginning of every frame, B-ACK is sent to all nodes in the network. In [11,12], a system was designed using a Markov chain whose state is the number of backlogged nodes, similar to [9,10]. However, the difference is that the number of backlogged nodes can be decreased by more than one because the state in [11,12] can be changed individually by frame, not by slot. Thus, there are many cases of the change of the number of backlogged nodes and the computational complexity of the state transition probability grows exponentially fast with the number of nodes. Because of high computational complexity, both Markov chain models [11,12] partially use simulation to obtain the state transition probability, instead of mathematical analysis. To the best of our knowledge, there are no fully mathematical models for analyzing stability, throughput, and packet delay when S-ALOHA uses a delayed ACK.

In this paper, we simply design a variant of S-ALOHA, namely Block ACK-Based Slotted ALOHA (BAS-ALOHA), by modifying BS-ALOHA [1]. Similar to BS-ALOHA, our BAS-ALOHA has a framed structure to estimate *PDC* periodically, and provides a time alignment mechanism to reduce guard time. The framed structure consists of a time period for beaconing and a group of multiple time slots for random access. The difference is that a beacon of BAS-ALOHA additionally includes B-ACK to report the results of all transmissions attempted in the previous frame at once. Thus, at the start of every frame, all the nodes that receive the beacon can recognize whether their transmission of the previous frame was successful. BAS-ALOHA can largely reduce retransmission overhead using B-ACK, instead of multiple immediate ACKs. However, to design BAS-ALOHA is not main contribution. As mentioned above, to the best of our knowledge, there are no fully mathematical models for analyzing stability, throughput, and packet delay when S-ALOHA uses a delayed ACK such as B-ACK. Therefore, the main contribution of this paper is to construct the fully mathematical models for BAS-ALOHA. First, the analytical model for stability is constructed based on [12] because of the similar retransmission method using B-ACK. Thus, the analytical model for BAS-ALOHA is also constructed using a Markov Chain whose state is defined as the number of backlogged nodes. However, as mentioned before, in order to obtain the state transition probability, simulation is used in [12]. On the other hand, we construct fully mathematical models that do not require any simulation. In addition, our mathematical models are not only for stability, but also for throughput and packet delay of BAS-ALOHA, whereas there are no analytical models for throughput and packet delay in [12]. We expect our analytical models to be a foundation for deriving fully mathematical models for variants of S-ALOHA using a framed structure and B-ACK.

The rest of this paper is organized as follows. In Section 2, we explain the proposed BAS-ALOHA. In Section 3, we first explain the traffic model and spatial distribution of nodes. Then, we derive fully mathematical models for the stability, throughput, and packet delay of BAS-ALOHA. Moreover, we compare the results with the simulation results to verify our mathematical models. In Section 4, we show numerical results using various parameters. Finally, we conclude the remarks in Section 5.

2. Block ACK-Based Slotted ALOHA (BAS-ALOHA)

We consider a single-channel long propagation network that consists of a coordinator and multiple nodes. We assume that all nodes and the coordinator are synchronized perfectly by GPS and that they transmit packets of equal size to the coordinator.

BAS-ALOHA is designed based on BS-ALOHA [1], and thus BAS-ALOHA also has a framed structure (Table 1) that starts with a beacon signal that contains the timestamp of the coordinator. After receiving it, all nodes estimate *PDC* using the timestamp. The difference from BS-ALOHA is that the BAS-ALOHA beacon additionally contains B-ACK that represents information for idle, collided, or successful slots in the previous frame. Because of B-ACK, each node can determine whether retransmission is required.

Table 1
Main notations for BAS-ALOHA framed structure.

Variable	Description
t_{start}	Start time of a frame
T_{frame}	Frame duration
T_B	Beacon transmission time
T_{MAX_RTT}	Maximum round-trip time
R	Maximum communication range
c	Signal propagation speed
t_γ	$t_{start} + T_B + T_{MAX\_RTT}$
γ	$T_B + T_{MAX\_RTT}$
$t_{s,\,j}$	Start time of the slot j
T_S	Slot length
M	Number of slots in a frame
t_i	Time at which a packet at ith node is generated
$t_{v,i}$	Virtual packet arrival time of ith node
$t_{B,\,i}$	Beacon arrival time of node i
τ_i	PDC from node i
$t_{tx,\,i}$	Time at which node i starts to transmit a packet
T_P	Transmission delay of a packet
T_G	Guard time length
T_I	Interval length
I_j	jth interval
N	Number of nodes in a network
v_{MAX}	Maximum node speed

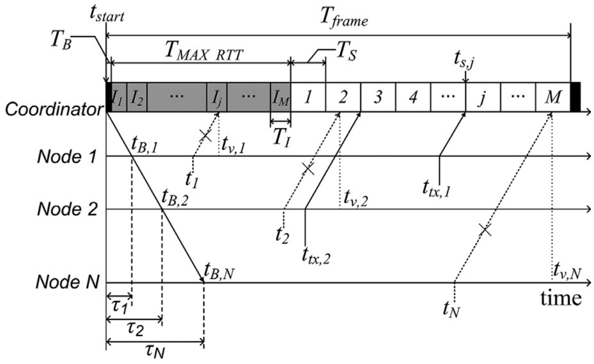


Fig. 1. *PDC* estimation and time alignment in a BAS-ALOHA frame.

2.1. PDC estimation and time alignment

We define several variables for the BAS-ALOHA framed structure, as shown in Fig. 1. Let t_{start}, T_{frame}, T_B, T_{MAX_RTT}, and $t_{s,\,j}$ be the start time of a frame, frame duration, beacon transmission time, maximum round-trip time of the maximum communication range R, and start time of the jth slot, respectively. T_{MAX_RTT} is given by $T_{MAX\_RTT} = 2R/c$, where c is the signal propagation speed. The framed structure consists of a coordinator time slot and a group of multiple time slots. The coordinator time slot whose guard time length is T_{MAX_RTT} is for beaconing. After the coordinator time slot of length $T_B + T_{MAX\_RTT}$, M slots of length T_S are available for random access. Including an initial delay of length, T_{MAX_RTT} allows the farthest node from the coordinator to access the first slot. We define t_i as the time at which a packet at the ith node is generated, and $t_{tx,\,i}$ as the time at which the ith node starts to transmit the packet. We also define $t_{v,i}$ as the virtual packet arrival time, which is the time at which a packet generated by the ith node would arrive if the packet were transmitted immediately after being generated. Thus, $t_{v,i}$ is obtained by $t_{v,i} = t_i + \tau_i$, where τ_i is the *PDC* of the ith node.

All nodes can estimate *PDC* in every frame using coordinator beaconing. As shown in Fig. 1, the coordinator broadcasts a beacon at t_{start}, the start of a frame, and the ith node receives it at $t_{B,\,i}$, the beacon arrival time. *PDC* of the ith node is then obtained by $\tau_i = t_{B,\,i} - t_{start}$. Using GPS and framed structure, periodic *PDC* estimation is possible.

Next, we explain a random access period where each node executes the time alignment mechanism to reduce guard time. In addition, we present a distribution method for packets generated during the time of beaconing. These packets are distributed evenly over the random access period.

When the ith node generates a packet, it determines the start time of transmission $t_{tx,\,i}$ based on $t_{v,i}$ and τ_i. First, it checks whether $t_{v,i}$ falls within the M slots available for transmission. For simplicity, let t_γ and γ be $t_{start} + T_B + T_{MAX\_RTT}$ and $T_B + T_{MAX\_RTT}$, respectively. If $t_{v,i}$ falls within the first $M-1$ slots, *i.e.*, $t_\gamma \le t_{v,i} \le t_\gamma + (M-1) \times T_S$, the node selects the next slot to start after $t_{v,i}$. If $t_{v,i}$ falls within slot M, *i.e.*, $t_\gamma + (M-1) \times T_S \le t_{v,i} \le t_\gamma + M \times T_S$, the packet is transmitted in the first slot of the next frame because there are no more slots available in the current frame. Therefore, if $t_\gamma \le t_{v,i} \le t_\gamma + M \times T_S$, index k of the selected slot is given by

$$k = \left\{ \left(\left\lceil \frac{t_{v,i} - t_\gamma}{T_S} \right\rceil \right) mod\ M \right\} + 1. \quad (1)$$

On the other hand, if $t_{v,i} < t_\gamma$, *i.e.*, the packet is generated during the time of beaconing, the node postpones transmission and selects the first slot for transmission. Then the collision probability of the first slot is different from that of the other slots. To equalize the collision probability of every slot, we uniformly divide γ into M intervals $(I_1, \ldots, I_M)$ and define T_I as the interval length obtained by $T_I = (T_B + T_{MAX\_RTT})/M = \gamma/M$. If $t_{v,i}$ is within I_k, the packet is transmitted in slot k obtained by

$$k = \left\lceil \frac{t_{v,i} - t_{start}}{T_I} \right\rceil. \quad (2)$$

In this way, BAS-ALOHA can evenly distribute packets generated during the time of beaconing over the random access period. Consequently, BAS-ALOHA can equalize the collision probability of every slot and provide fair packet delay by allowing k to be proportional to $t_{v,i}$.

After selecting the slot, the node performs time alignment in order to reduce guard time while resolving space-time uncertainty by adjusting the transmission start time so that the packet arrives at the beginning of a slot. If slot k is selected, $t_{tx,\,i}$ is obtained by $t_{tx,i} = t_{s,k} - \tau_i$.

For example, in Fig. 1, $t_{v,2}$ and $t_{v,N}$ are within slot 2 and M, respectively. Thus, node 2 selects slot 3. However, in the case of node N, there are no more slots after slot M in the frame. Thus, node N selects slot 1 of the next frame. Unlike node 2 and N, node 1 selects slot j because $t_{v,1}$ is within I_j. After selecting the slots for packet transmission, each node adjusts the transmission start time so that the packet arrives at the beginning of a slot. Thus, $t_{tx,\,1}$, $t_{tx,\,2}$, and $t_{tx,\,N}$ can be obtained by $t_{s,j} - \tau_1$, $t_{s,3} - \tau_2$, and $t_{s,1} - \tau_N$, respectively.

2.2. Retransmission

After receiving a beacon, all nodes can update *PDC* for time alignment and determine whether their transmission in the previous frame is successful using B-ACK. We assume that all nodes transmit a packet in a frame.

If a node fails in transmission, it becomes backlogged. The backlogged node attempts to retransmit the packet in the next frame with a certain probability p_1. If the probability p_1 is satisfied, the backlogged node determines a slot among M slots with uniform distribution for retransmission. If the probability p_1 is not satisfied, the backlogged node reattempts transmission in the next frame.

After determining a slot for retransmission, the backlogged node performs a time alignment mechanism with the updated *PDC* at the start of this frame. If node i updates τ_i and determines slot j for retransmission, $t_{tx,\,i}$ is obtained by $t_{tx,i} = t_{s,j} - \tau_i$.

3. Performance analysis of BAS-ALOHA

In this section, we analyze the performance of BAS-ALOHA (Table 2). Before the performance analysis, we explain the traffic model and spatial distribution of nodes. Then we derive mathematical models for stability, throughput, and packet delay for BAS-ALOHA. Unlike past analytical models, we provide fully mathematical models for analyzing stability and throughput. Furthermore, we provide a new mathematical model for analyzing packet delay.

Table 2
Main notations for performance analysis.

Variable	Description
G	Total load of a network
G_F	Load from all fresh nodes
G_B	Load from all backlogged nodes
x_F	Number of fresh nodes
x_B	Number of backlogged nodes
$X_{F,\,i}$	Number of fresh nodes in frame i
$X_{B,\,i}$	Number of backlogged nodes in frame i
p_0	Prob. that a fresh node transmits a packet in frame
p_1	Prob. that a backlogged node retransmits a packet in frame
F_S	Number of fresh nodes that successfully transmit
F_U	Number of fresh nodes that attempt transmission, but are unsuccessful
B_S	is Number of backlogged nodes that successfully transmit
B_U	Number of backlogged nodes that attempt transmission, but are unsuccessful
r	Number of fresh nodes that attempt transmission
t	Number of backlogged nodes that attempt transmission
u	Number of successful nodes
L	Number of nodes that attempt transmission
$\boldsymbol{\Omega_T}$	System state when L nodes attempt transmission
$\boldsymbol{Z_L}$	Set in which all possible Ω_T are elements
$\omega_{T,\,i}$	Number of slots where i among L nodes attempt transmission
ρ	Ratio of the number of successful nodes to the number of nodes that attempt transmission

3.1. Traffic model and spatial distribution of nodes

In the BAS-ALOHA system, there are N nodes in total, each of which resides either in a fresh (F) or backlogged (B) state. In the beginning, all nodes are in state F. Every node in state F attempts to transmit a new packet in the current frame with probability p_0. All nodes recognize success or failure of their transmission after receiving a beacon at the start of the next frame. If transmission is successful, the node remains in state F. On the other hand, if a node fails in transmission, it enters state B and attempts retransmission of the lost packet with probability p_1. If the retransmission is successful, the node returns to state F; otherwise, the node remains in state B. Nodes in state B do not generate new traffic.

Fig. 2 shows the traffic model for BAS-ALOHA. Nodes in states F and B generate load G_F and G_B, respectively. Thus, total load G of the system is the sum of G_F and G_B.

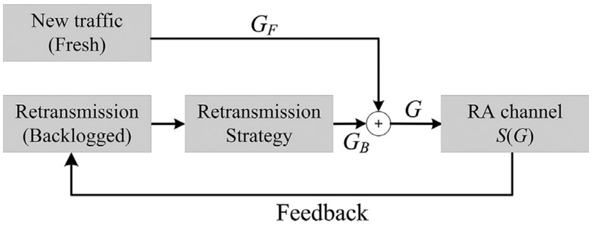


Fig. 2. Traffic model for BAS-ALOHA.

We assume that all nodes are distributed in a circle whose radius is *R*. A coordinator is located at the center of the circle. We also assume that the distance between the coordinator and a node is a random variable that is uniformly distributed from zero to *R*.

3.2. Stability analysis

Many studies [9,11–13] have already considered the stability of S-ALOHA systems. In [12], stability was defined as the ability of a system to maintain equilibrium or return to the initial state after experiencing a distortion. We also use the same definition presented in [12] for stability.

To show stability, drift analysis, which shows the change of the number of backlogged nodes, was used in [11,12]. However, they did not provide fully mathematical models for drift analysis. Instead, they used simulation partially.

BAS-ALOHA can be modeled as a discrete-time Markov chain. Let $X_{F,\,i}$ and $X_{B,\,i}$ denote the number of nodes in states *F* and *B* in the beginning of *i*th frame, respectively. The discrete-time Markov chain can be described either by $X_{F,\,i}$ or $X_{B,\,i}$ because both variables are connected by $X_{F,i} = N - X_{B,i}$. Thus, we use $X_{B,\,i}$ as the Markov state variable. In the beginning, $X_{B,\,0}$ is zero. We define state transition probability $P(X_{B,i+1} = x' | X_{B,i} = x)$, which is simply denoted by $P(x'|x)$, as the probability to move within one frame from backlog state *x* to state *x*′. In BAS-ALOHA, the state transition appears individually by frame, and the number of nodes in state *B* can decrease by more than one. As mentioned in [12], this is the major difference to S-ALOHA [9], where the state transition appears individually by slot, and the number of nodes in state *B* can only decrease by one. Fig. 3 shows the state transition diagram consisting of four states. In each state, there is a number which denotes the number of backlogged nodes. In particular, the state *K* is a representative state for states whose number is from 2 to $N-1$. The state transition from state 0 to state 1 is impossible because at least the collision occurs by two nodes.

Let F_S denote the number of fresh nodes that successfully transmit in frame *i*. Let F_U denote the number of fresh nodes who attempt a transmission, but are unsuccessful in frame *i*. Similarly, B_S is the number of backlogged nodes who attempt retransmission and are successful, and B_U is the number of backlogged nodes who attempt retransmission, but are unsuccessful in frame *i*. Thus, in frame *i*, $r = F_S + F_U$, $t = B_S + B_U$, and $u = F_S + B_S$, where *r* is the total number of fresh nodes that attempt transmission, *t* is the total number of backlogged nodes that attempt transmission, and *u* is the total number of successful nodes, respectively.

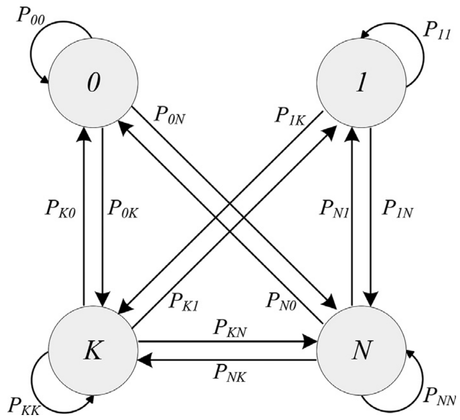


Fig. 3. State transition diagram with *N* nodes and $P_{xx'}$ denotes $P(x'|x)$ and *K* is any integer for $1 < K < N$.

Let $P(r,\,t,\,u|x_B)$ be the conditional probability of *r* fresh nodes and *t* backlogged nodes attempting transmission, and *u* nodes among them being successful in a frame given that the number of backlogged nodes is x_B. The joint probability is given by

$$\begin{aligned} P(r,t,u|x_B) &= \binom{x_F}{r} p_0^r (1-p_0)^{x_F} \binom{x_B}{t} p_1^t (1-p_1)^{x_B - t} \\ &\quad \times q_M(r+t,u) \\ &= \binom{N-x_B}{r} p_0^r (1-p_0)^{N-x_B-r} \\ &\quad \times \binom{x_B}{t} p_1^t (1-p_1)^{x_B-t} \times q_M(r+t,u) \end{aligned} \tag{3}$$

where $q_M(r+t,u)$ is the probability of $r+t$ nodes attempting transmission and *u* nodes among them being successful when the number of slots in a frame is *M*. If the number of backlogged nodes is x'_B in the $(i+1)$th frame, the difference between x'_B and x_B can be obtained by $F_U - B_S$ and also be reformulated into

$$u = r + x_B - x'_B. \tag{4}$$

The state transition probability $P(x'_B|x_B)$ can be formulated by combining (3) and (4) into (5),

$$\begin{aligned} P(x'_B|x_B) &= \sum_{r,t} P(r,t,u|x_B) \\ &= \sum_{r,t} P(r,t,r+x_B-x'_B|x_B) \\ &= \sum_{r=0}^{N-x_B} \sum_{t=0}^{x_B} \binom{N-x_B}{r} p_0^r (1-p_0)^{N-x_B-r} \\ &\quad \times \binom{x_B}{t} p_1^t (1-p_1)^{x_B-t} \times q_M(r+t, r+x_B-x'_B). \end{aligned} \tag{5}$$

Because of the high computational cost of the state transition probability for all cases, drift analysis was used in [11,12]. We also use drift $d(x_B)$, which is the expectation value of the change of the backlog state $X_{B,\, i}$ individually by frame. We can obtain drift $d(x_B)$ by

$$d(x_B) = E[X_{B,i+1} - X_{B,i} | X_{B,i}] = \sum_{x'_B} (x'_B - x_B) P(x'_B | x_B). \quad (6)$$

By combining (6) and , $d(x_B)$ can be expressed as

$$\begin{aligned} d(x_B) &= \sum_{r,t,u} (r-u) P(x'_B | x_B) \\ &= \sum_{r,t,u} (r-u) \sum_{r,t} P(r,t,u|x_B) \\ &= \sum_{r,t,u} (r-u) P(r,t,u|x_B) \\ &= E[R(x_B)] - E[U(x_B)] \quad (7) \end{aligned}$$

where $E[R(x_B)]$ and $E[U(x_B)]$ are the average number of fresh and successful nodes, respectively, when the number of backlogged nodes is x_B. We can obtain $E[R(x_B)]$ by

$$E[R(x_B)] = (N - x_B) p_0 \quad (8)$$

and also obtain $E[U(x_B)]$ by

$$\begin{aligned} E[U(x_B)] &= \sum_{r,t,u} u P(r,t,u|x_B) \\ &= \sum_{u=0}^{N} u \sum_{r=max(0,u-x_B)}^{N-x_B} \sum_{t=0}^{x_B} P(r,t,u|x_B) \\ &= \sum_{u=0}^{N} u \sum_{r=max(0,u-x_B)}^{N-x_B} \sum_{t=0}^{x_B} \binom{N-x_B}{r} \\ &\times p_0^r (1-p_0)^{N-x_B-r} \binom{x_B}{t} p_1^t (1-p_1)^{x_B-t} \\ &\times q_M(r+t, u). \quad (9) \end{aligned}$$

For drift analysis, factor $q_M(\,\cdot\,)$ is important. Moreover, it is also used in throughput and delay analysis. The problem is the calculation of $q_M(\,\cdot\,)$. In [11,12], no mathematical models for obtaining $q_M(\,\cdot\,)$ were provided. Instead, simulation was used to calculate $q_M(\,\cdot\,)$. Therefore, we focused on deriving a mathematical formula for $q_M(\,\cdot\,)$.

Let L and $\boldsymbol{\Omega_L}$ denote the number of nodes that attempt transmission and the *system state* when L nodes attempt transmission, respectively. Thus, L is $r+t$, and $\boldsymbol{\Omega_L}$ is expressed as

$$\boldsymbol{\Omega_L} = (\omega_{L,1}, \omega_{L,2}, \omega_{L,3}, \ldots, \omega_{L,L}) \quad (10)$$

where $\omega_{L,\, i}$ is the number of slots where i nodes among L nodes attempt transmission. For example, system state $\boldsymbol{\Omega_L}$ = (2,3,1,0,0) means that there are two slots where one node attempts transmission, three slots where two nodes attempt transmission, and one slot where three nodes attempt transmission. Given $\boldsymbol{\Omega_L}$, the number of successful packets is $\omega_{L,\, 1}$, whereas the number of collided packets is $L - \omega_{L,1}$. In addition, L can be obtained by

$$L = \sum_{i=1}^{L} i \times \omega_{L,i}, \quad (11)$$

when system state $\boldsymbol{\Omega_L}$, not r and t, is given. Let m denote the number of slots occupied by a node or nodes. The value for m can be obtained by

$$m = \sum_{i=1}^{L} \omega_{L,i} \le min(M, L). \quad (12)$$

The value for m cannot be greater than either frame length M or the number of nodes transmitting in frame L.

Algorithm 1: Obtaining $\boldsymbol{Z_L}$.

```
1  Input:M,L
2  Initialization: T = 1, ω_{1,1} = 1, Ω_1 = (ω_{1,1}), Z_1 = {Ω_1}
3
4  for T = 2; T = L; T++
5      Z_T = φ
6      for all Ω_{T−1} ∈ Z_{T−1}
7          for i = 1; i ≤ T − 1; i++
8              ω_{T,i} = ω_{T−1,i}
9          end for
10         ω_{T,T} = 0
11
12 //case 1: Tth transmission is successful
13         ω_{T,1} = ω_{T,1} + 1
14         Ω_T ← (ω_{T,1}, ω_{T,2}, ··· , ω_{T,T})
15         calculate m with Ω_T by (12)
16         if Ω_T ∉ Z_T and m ≤ min(M, L)
17             Z_T = Z_T ∪ {Ω_T}
18         end if
19         ω_{T,1} = ω_{T,1} − 1
20         Ω_T ← (ω_{T,1}, ω_{T,2}, ··· , ω_{T,T})
21
22 //case 2: Tth transmission is not successful
23         for i = 1; i ≤ T − 1; i++
24             if ω_{T,i} ≥ 1
25                 ω_{T,i} = ω_{T,i} − 1
26                 ω_{T,i+1} = ω_{T,i+1} + 1
27                 Ω_T ← (ω_{T,1}, ω_{T,2}, ··· , ω_{T,T})
28                 calculate m with Ω_T by (12)
29                 if Ω_T ∉ Z_T and m ≤ min(M, L)
30                     Z_T = Z_T ∪ {Ω_T}
31                 end if
32                 ω_{T,i} = ω_{T,i} + 1
33                 ω_{T,i+1} = ω_{T,i+1} − 1
34             end if
35         end for
36     end for
37 end for
```

There are several states where L nodes attempt transmission. As mentioned before, $\boldsymbol{\Omega_L}$ means *system state* when L nodes attempt transmission. Let $\boldsymbol{Z_L}$ be the set where all $\boldsymbol{\Omega_L} \in \boldsymbol{Z_L}$. In order to obtain $\boldsymbol{Z_L}$ with the given L and M, we propose Algorithm 1. As indicated in line 2, this algorithm initializes system state $\boldsymbol{\Omega_1}$ and set $\boldsymbol{Z_1}$ when the number of transmission attempts, denoted as T, is one. Then, as indicated in lines 4 to 37, state $\boldsymbol{\Omega_T}$ and set $\boldsymbol{Z_T}$ are obtained iteratively, whereas the value for T ranges from 2 to L. There are two cases in the iteration. First, as indicated from lines 12 to 20, the Tth transmission is successful. The value for $\omega_{T,1}$ can increase by one, state $\boldsymbol{\Omega_T}$ is updated, and m is calculated. Subsequently, if the same state does not exist in $\boldsymbol{Z_T}$, state $\boldsymbol{\Omega_T}$ is added to set $\boldsymbol{Z_T}$. Then, state $\boldsymbol{\Omega_T}$ is returned to the original state as indicated in line 19. Second, as indicated from lines 22 to 35, the Tth transmission is not successful. This means that any existing transmissions, *i.e.*, all $\omega_{T,i}$ for $\omega_{T,i} \geq 1$, and the Tth transmission collide. Thus, as indicated in lines 25 and 26, $\omega_{T,i+1}$ increases by one, whereas $\omega_{T,i}$ decreases by one. Subsequently, state $\boldsymbol{\Omega_T}$ is updated and m is calculated. If the same state does not exist in $\boldsymbol{Z_T}$, state $\boldsymbol{\Omega_T}$ is added to set $\boldsymbol{Z_T}$. Then, state $\boldsymbol{\Omega_T}$ is returned to the original state as indicated in lines 32 and 33. After finishing the Algorithm 1, $\boldsymbol{Z_L}$ is obtained.

With the $\boldsymbol{Z_L}$, we can obtain $q_M(L, u)$. The value for $q_M(L, u)$ can be obtained by

$$q_M(L,u) = \sum_{\boldsymbol{\Omega_L}} Pr(\omega_{L,1} = u|\boldsymbol{\Omega_L}) \times P(\boldsymbol{\Omega_L}) \quad (13)$$

where $Pr(\omega_{L,1} = u|\boldsymbol{\Omega_L})$ is the conditional probability of u nodes succeeding in transmission given that the system state is $\boldsymbol{\Omega_L}$. In addition, $P(\boldsymbol{\Omega_L})$ is the probability of the system state being $\boldsymbol{\Omega_L}$. Therefore, $q_M(L, u)$ can simply be rewritten by

$$q_M(L,u) = \sum_{\boldsymbol{\Omega_L} \in \boldsymbol{Z_L}(u)} P(\boldsymbol{\Omega_L}) \quad (14)$$

where $\boldsymbol{Z_L}(u)$ is the set of states for which $\omega_{L,1} = u$. In [14], $P(\boldsymbol{\Omega_L})$ is given by

$$P(\boldsymbol{\Omega_L}) = \frac{M!L!}{M^L(M-m)!\prod_{i=1}^{L}(i!)^{\omega_{L,i}}\omega_{L,i}!}. \quad (15)$$

Now we obtain the fully mathematical equation for $q_M(L, u)$.

By combining (9) and (14), we can rewrite $E[U(x_B)]$ as

$$\begin{aligned} E[U(x_B)] &= \sum_{u=0}^{N} u \sum_{r=max(0,u-x_B)}^{N-x_B} \sum_{t=0}^{x_B} \binom{N-x_B}{r} \\ &\times p_0^r(1-p_0)^{N-x_B-r} \binom{x_B}{t} p_1^t(1-p_1)^{x_B-t} \\ &\times q_M(r+t,u) \\ &= \sum_{u=0}^{N} u \sum_{r=max(0u-x_B)}^{N-x_B} \sum_{t=0}^{x_B} \binom{N-x_B}{r} \\ &\times p_0^r(1-p_0)^{N-x_B-r} \binom{x_B}{t} p_1^t(1-p_1)^{x_B-t} \\ &\times \sum_{\boldsymbol{\Omega_{r+t}} \in \boldsymbol{Z(u)}} P(\boldsymbol{\Omega_{r+t}}). \end{aligned} \quad (16)$$

Finally, we can obtain $d(x_B)$ in (7) using (8) and (16).

3.3. Throughput analysis

In BAS-ALOHA, a slot whose length is T_S is divided into two parts: T_P and T_G. Part T_G is an additional guard time in a slot to compensate for the drift of the estimated *PDC* caused by the high mobility of a node. More precisely, the estimated *PDC* for transmission is estimated after receiving a beacon, not before transmission. As a result, *PDC* is inaccurate and causes unexpected collisions. Thus, BAS-ALOHA employs additional guard time T_G in a slot in order to prevent collisions caused by an inaccurate *PDC* at the coordinator. The amount of drift of the estimated *PDC* can be greatest when two nodes move in opposite directions with maximum velocity during T_{frame}. Thus, T_G is set as

$$T_G = 2v_{MAX} \times \frac{T_{frame}}{c} \quad (17)$$

where c is the signal propagation speed of 3×10^8 m/s, and v_{MAX} is the maximum node speed. The equation for T_{frame} can be written as

$$T_{frame} = T_B + T_{MAX\_RTT} + M(T_P + T_G) \quad (18)$$

and we replace T_{frame} in (17) with (18) because T_{frame} contains T_G recursively. Then we can obtain T_G by

$$T_G = \frac{2v_{MAX}(T_B + T_{MAX\_RTT} + MT_P)}{c - 2Mv_{MAX}}. \quad (19)$$

We define normalized throughput as the ratio of the total time required for successful packet transmission to frame length. The normalized throughput for BAS-ALOHA is dependent on x_B because $U(x_B)$, which also depends on x_B, is needed to obtain it. Thus, we denote $S(x_B)$ as the normalized throughput, and $S(x_B)$ can be expressed as

$$S(x_B) = \frac{E[U(x_B)] \times T_P}{T_{frame}} \quad (20)$$

where $E[U(x_B)]$ is given by (16).

3.4. Packet delay analysis

We define packet delay D as the duration from a packet being generated to the coordinator receiving the packet successfully in equilibrium. Packet delay D depends on the number of transmissions until successful transmission. Therefore, the expected packet delay, denoted by $E[D]$, can be obtained by

$$E[D] = \sum_{i=1}^{\infty} E[D_i] \times P_S(i) \quad (21)$$

where $E[D_i]$ is the expected packet delay when the coordinator successfully receives a packet at the ith transmission and $P_S(i)$ is the probability of the ith transmission being successful. Probability $P_S(i)$ is a geometric distribution because transmission is performed until success is achieved. Thus, $P_S(i)$ can be expressed as

$$P_S(i) = (1-\rho)^{i-1} \times \rho \quad (22)$$

where ρ is the probability of the transmission being successful. In other words, ρ is the ratio of the number of successful nodes to the number of nodes that attempt transmission. Thus, ρ can be expressed as

$$\rho = \frac{E[U(\hat{x_B})]}{(N-\hat{x_B}) \times p_0 + \hat{x_B} \times p_1} \quad (23)$$

where $\hat{x_B}$ is the number of backlogged nodes in equilibrium.

In BAS-ALOHA, a frame consists of a beacon period (*BC*) and random access period (*RA*). As shown in Fig. 1, a length for BC is $T_B + T_{MAX\_RTT}$, *i.e.*, γ, and a length for RA is $T_S \times M$.

In the first transmission, the slot decision depends on the virtual packet arrival time, whereas the slot for retransmission is determined using uniform distribution with a range from one to M. Therefore, $E[D_1]$ is completely different from the other $E[D_i]$ for $i \geq 2$, and (21) can be rewritten as

$$E[D] = E[D_1] \times P_S(1) + \sum_{i=2}^{\infty} E[D_i] \times P_S(i). \tag{24}$$

For the first transmission, $E[D_1]$ can be expressed as

$$E[D_1] = E[D_1|X = BC] \times P(X = BC) + E[D_1|X = RA] \times P(X = RA) \tag{25}$$

where $E[D_1|X = BC]$ and $E[D_1|X = RA]$ are the expected packet delay when the virtual packet arrival time is within BC and RA, respectively, and the first transmission is successful. Probabilities $P(X = BC)$ and $P(X = RA)$ are the probabilities of the virtual packet arrival time being within BC and RA, respectively. Both $P(X = BC)$ and $P(X = RA)$ are given by

$$P(X = BC) = \frac{\gamma}{T_{frame}}$$
$$P(X = RA) = 1 - P(X = BC). \tag{26}$$

In (25), $E[D_1|X = BC]$ can be expressed as

$$E[D_1|X = BC] = \sum_{k=1}^{M} D_{BC,k} \times P(interval = k|X = BC) \tag{27}$$

where $D_{BC,\,k}$ is the expected packet delay when the coordinator receives the packet successfully in slot k, and $P(interval = k|X = BC)$ is the probability that the virtual packet arrival time is within interval I_k, when the virtual packet arrival time is also within BC. This probability can be obtained by

$$P(interval = k|X = BC) = \frac{\frac{T_I}{\gamma} \times \frac{\gamma}{T_{frame}}}{\frac{\gamma}{T_{frame}}} = \frac{T_I}{\gamma}. \tag{28}$$

Also, the $D_{BC,\,k}$ is expressed as

$$D_{BC,k} = \frac{T_I}{2} + (M - k)T_I + (k - 1)T_S + E_p. \tag{29}$$

The $D_{BC,\,k}$ is divided into four parts. First, $T_I/2$ is the average amount of time from virtual packet arrival time to the end of interval I_k. Then, there are still $M - k$ intervals to the end of BC. Thus, the second part is $(M - k)T_I$. The third part is $(k - 1)T_S$, which is the amount of time from the start of the first slot in RA to the start of slot k. In the last part, E_p is the expected PDC. By using (28) and (29), we can rewrite (27) to

$$E[D_1|X = BC] = \sum_{k=1}^{M} \left\{ \frac{T_I}{2} + (M - k)T_I + (k - 1)T_S + E_p \right\} \times \frac{T_I}{\gamma}. \tag{30}$$

In (25), $E[D_1|X = RA]$ can be expressed as

$$E[D_1|X = RA] = \sum_{k=1}^{M} D_{RA,k} \times P(slot = k|X = RA) \tag{31}$$

where $D_{RA,\,k}$ is the expected packet delay when the coordinator receives the packet successfully in slot k, and $P(slot = k|X = RA)$ is the probability that the virtual packet arrival time is within slot k, when the virtual packet arrival time is also within RA. This probability can be obtained by

$$P(slot = k|X = RA) = \frac{\frac{T_S}{T_{frame} - \gamma} \times \frac{T_{frame} - \gamma}{T_{frame}}}{\frac{T_{frame} - \gamma}{T_{frame}}} = \frac{T_S}{T_{frame} - \gamma}. \tag{32}$$

For $D_{RA,\,k}$, there are two cases. If $k < M$, $D_{RA,\,k}$ can be obtained by

$$D_{RA,k} = \frac{T_S}{2} + E_p \tag{33}$$

where $T_S/2$ means the average amount of time from virtual packet arrival time to the end of slot $k - 1$. However, if $k = M$, the transmission is delayed to the first slot of the next frame. Thus, $D_{RA,\,M}$ can be obtained by

$$D_{RA,M} = \frac{T_S}{2} + \gamma + E_p \tag{34}$$

where γ becomes the additional delay to the first slot of the next frame. Thus, using (33) and (34), $\sum_{k=1}^{M} D_{RA,k}$ can be obtained by

$$\sum_{k=1}^{M} D_{RA,k} = M\left(\frac{T_S}{2} + E_p\right) + \gamma. \tag{35}$$

Using (32) and (35), we can rewrite (31) as

$$E[D_1|X = RA] = \left\{ M\left(\frac{T_S}{2} + E_p\right) + \gamma \right\} \times \frac{T_S}{T_{frame} - \gamma}. \tag{36}$$

Finally, using (26), (30), and (36), we can rewrite $E[D_1]$ in (25) as

$$E[D_1] = \sum_{k=1}^{M} \left\{ \frac{T_I}{2} + (M - k)T_I + (k - 1)T_S + E_p \right\} \times \frac{T_I}{T_{frame}} + \left\{ M\left(\frac{T_S}{2} + E_p\right) + \gamma \right\} \times \frac{T_S}{T_{frame}}. \tag{37}$$

During retransmission, the slot is determined randomly using uniform distribution with the range from one to M, unlike the first transmission. As demonstrated in (24), $E[D_i]$ for $i \geq 2$ is completely different from $E[D_1]$. For $i \geq 2$, $E[D_i]$ can be obtained by

$$E[D_i] = \frac{T_{frame}}{2} + (i - 2)T_{frame} + \gamma + \left.\frac{T_{frame} - \gamma}{2}\right) \tag{38}$$

where the first term $T_{frame}/2$ is the average amount of time between the first transmission and the beacon arrival, which notifies that transmission has failed. Then, during $(i - 2)$ frames, retransmissions fail and the corresponding delay is the second term $(i - 2)T_{frame}$. In the last transmission, which is eventually successful, there are two delays: γ for beaconing and $(T_{frame} - \gamma)/2$ for the average packet delay from the first slot to a certain slot used for the ith transmission. These are the third and last term, respectively. We can briefly rewrite (38) as

$$E[D_i] = (i-1)T_{frame} + \frac{\gamma}{2}. \tag{39}$$

The summation of $E[D_i]$ for $2 \le i \le \infty$ is included in (24). This can be obtained by Appendix A, and it is expressed as

$$\sum_{i=2}^{\infty} E[D_i] \times P_S(i) = T_{frame}\left(\frac{1}{\rho} - 1\right) + \frac{\gamma(1-\rho)}{2}. \tag{40}$$

Finally, $E[D]$ in (21) can be rewritten as

$$E[D] = E[D_1] \times P_S(1) + T_{frame}\left(\frac{1}{\rho} - 1\right) + \frac{\gamma(1-\rho)}{2}. \tag{41}$$

4. Numerical results

In this section, we show numerical results with various parameters: M, T_P, p_0, and p_1.

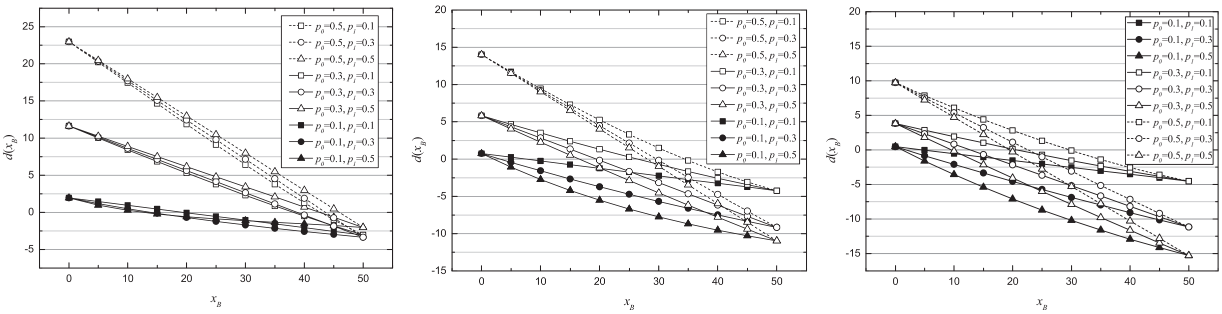
We consider the same environment as in [1,15]. Thus, we use almost the same parameters used in the studies. We set R, c, and v_{MAX} at 300 Nautical Miles (NMi), where 1 NMi = 1.852 km, 3×10^8 m/s, and 680 m/s, respectively. Thus, T_{MAX_RTT} is 3.704 ms, where $T_{MAX\_RTT} = 2R/c$. The value of N is set to 50. We consider two types of T_Ps: one whose transmission time is longer than the maximum propagation delay (3.0 ms), and another whose transmission time is shorter (0.1 ms).

Fig. 4 shows drift $d(x_B)$ according to x_B with various M, T_P, p_0, and p_1. Fig. 4(a), (b), and (c) show the drift when M is 10, 30, and 50, respectively. When x_B is zero, $d(x_B)$ is only affected by p_0, and thus BAS-ALOHA having the same p_0 provides the same $d(x_B)$, regardless of p_1. Reversely, when x_B is 50, $d(x_B)$ is only affected by p_1, and thus BAS-ALOHA having the same p_1 provides the same $d(x_B)$, regardless of p_0. In Fig. 4, $d(x_B)$ decreases, whereas x_B increases. In this case, $d(x_B)$ of zero means that the system is stable. Thus, the stable point is important to show the stability of the BAS-ALOHA system. In Fig. 4(a), x_B is in the ranges of 42 to 46, 39 to 42, and 14 to 20 at the stable point where p_0 is 0.5, 0.3, and 0.1, respectively. This means that BAS-ALOHA can support more backlogged nodes under the stable state, when p_0 is large. This can also be shown in Fig. 4(b) and (c). In Fig. 4(b), x_B is in the ranges of 27 to 35, 16 to 26, and 2 to 7 at the stable point where p_0 is 0.5, 0.3, and 0.1, respectively. In Fig. 4(c), x_B is in the ranges of 20 to 30, 10 to 20, and 1 to 4 at the stable point where p_0 is 0.5, 0.3, and 0.1, respectively. In most of the cases in Fig. 4, when p_0 is the same, BAS-ALOHA having a larger p_1 can provide more backlogged nodes at the stable point.

Fig. 5 shows the normalized throughput S according to the number of backlogged nodes. Fig. 5(a), (b), and (c) show S when M is 10, 30, and 50, respectively, and T_P is 0.1 ms, whereas Fig. 5(d), (e), and (f) show S when M is 10, 30, and 50, respectively, and T_P is 3.0 ms. When M and T_P are large, the maximum S is also high because the ratio of overhead γ to T_{frame} is relatively small. In all cases where p_0 and p_1 are the same, S is not affected by x_B because the total traffic load is not changed. When x_B is zero, S is only affected by p_0, and thus BAS-ALOHA having the same p_0 provides the same S, regardless of p_1. Reversely, when x_B is 50, S is only affected by p_1, and thus BAS-ALOHA having the same p_1 provides the same S, regardless of p_0. BAS-ALOHA appears similar to S-ALOHA, which provides maximum S when the total traffic load is one, having an overhead of γ in a frame. Therefore, the maximum S for BAS-ALOHA occurs at that total traffic load, *i.e.*, $(M - x_B)p_0 + x_B p_1$ is $M\gamma/T_{frame}$. Thus, when M is 30 or 50, the maximum S does not occur because the total traffic load with the given parameters (x_B, p_0, p_1) is always smaller than $M\gamma/T_{frame}$.

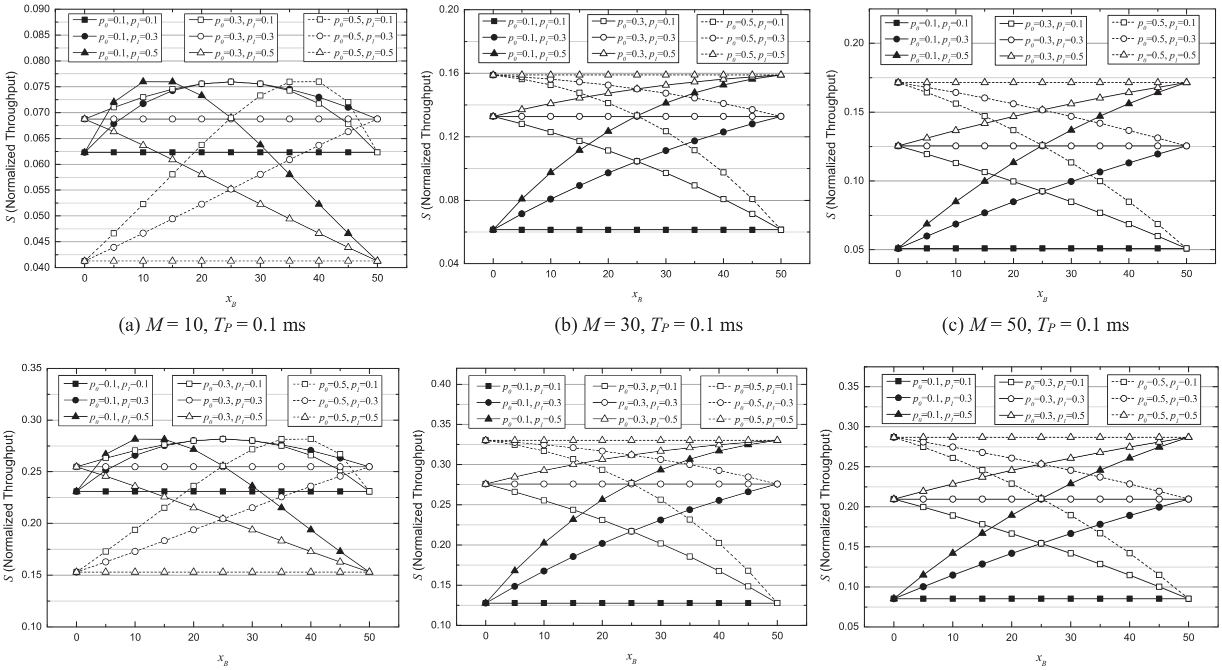
Fig. 6 shows expected packet delay D according to the number of backlogged nodes. Fig. 6(a), (b), and (c) show D when M is 10, 30, and 50, respectively, and T_P is 0.1 ms, whereas Fig. 6(d), (e), and (f) show D when M is 10, 30, and 50, respectively, and T_P is 3.0 ms. With the same T_P, D is large when M is small, because the number of collisions increases given the small M. With the same M, D is large when T_P is large. In all cases where p_0 and p_1 are the same, D is not affected by x_B because the total traffic load is not changed. When x_B is zero, D is only affected by p_0, and thus BAS-ALOHA having the same p_0 provides the same S, regardless of p_1. Reversely, when x_B is 50, D is only affected by p_1, and thus BAS-ALOHA having the same p_1 provides the same D, regardless of p_0.

To verify our analytical models, we compare the results from performance analysis with those from simulation. We develop two simulators to evaluate drift, throughput and packet delay.



(a) $M = 10$ (b) $M = 30$ (c) $M = 50$

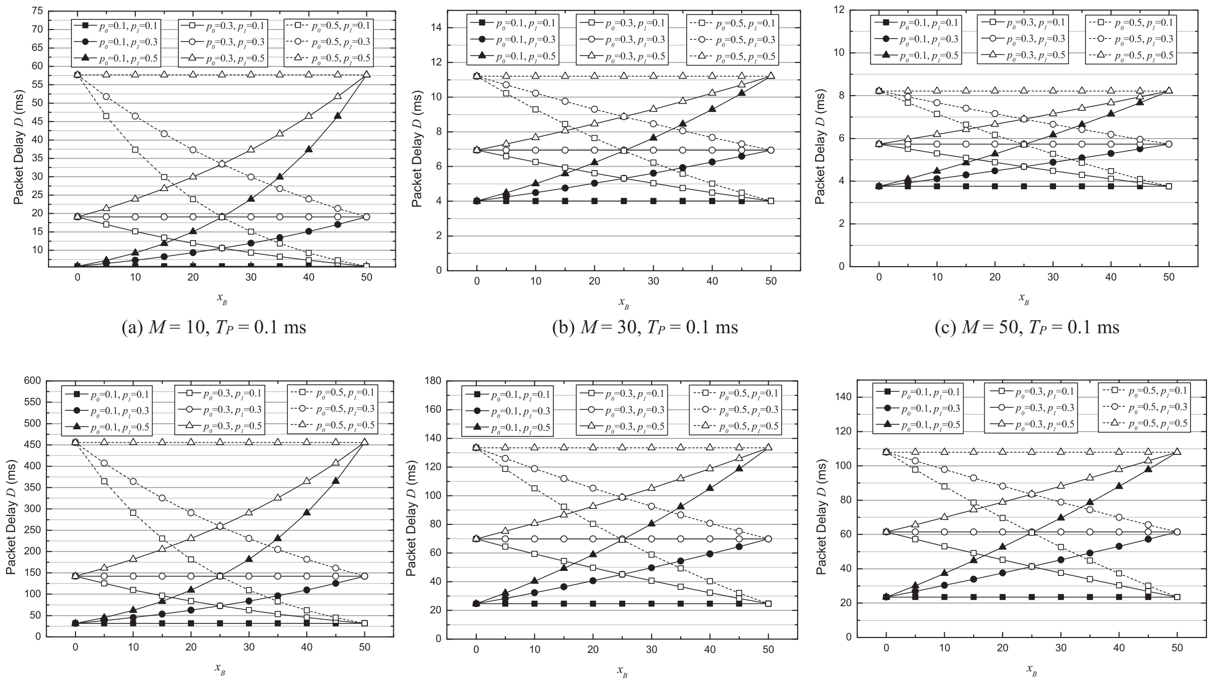
Fig. 4. Drift according to number of backlogged nodes (N=50).



(a) $M = 10$, $T_P = 0.1$ ms (b) $M = 30$, $T_P = 0.1$ ms (c) $M = 50$, $T_P = 0.1$ ms

(d) $M = 10$, $T_P = 3.0$ ms (e) $M = 30$, $T_P = 3.0$ ms (f) $M = 50$, $T_P = 3.0$ ms

Fig. 5. Normalized throughput according to number of backlogged nodes (N=50).



(a) $M = 10$, $T_P = 0.1$ ms (b) $M = 30$, $T_P = 0.1$ ms (c) $M = 50$, $T_P = 0.1$ ms

(d) $M = 10$, $T_P = 3.0$ ms (e) $M = 30$, $T_P = 3.0$ ms (f) $M = 50$, $T_P = 3.0$ ms

Fig. 6. Expected packet delay according to number of backlogged nodes in equilibrium (N=50).

First, Algorithm 2 shows the algorithm of the first simulation to obtain drift and throughput. The input variables use the same notations as in performance analysis. In particular, slots[M] is an array to present the number of packets transmitted in slot M. For example, if the value of slots[i] is three, three packets are transmitted in the ith slot and they cause collision. If the value of slots[i] is one, one packet is transmitted in the ith slot and successful. From line 3 to 5, slots[M] array is initialized. In line 7, n_pkt and n_pket_fr means the number of pack-

Algorithm 2: Obtaining drift and throughput.

```
1  Input:N, p0, p1, T_frame, M, T_P, x_B, t_s,i, slots[M]
2  // Slot state initialization
3  for i = 0; i <= M; i++
4      slots[i] = 0
5  end for
6
7  n_pkt = 0, n_pkt_fr = 0
8  for i = 0; i <= N; i++
9      if i > x_B // fresh node
10         if rand() <= p0 // packet generation
11             index = getSlotIndex(t_s,i)
12             slots[index]++
13             n_pkt++
14             n_pkt_fr++
15         end if
16     else // backlogged node
17         if rand() <= p1
18             index = getRandNum(M)
19             slots[index]++
20             n_pkt++
21         end if
22     end if
23 end for
24
25 n_pkt_suc = 0
26 for i = 1; i <= M; i++
27     if slots[i] == 1
28         n_pkt_suc++
29     end if
30 end for
31
32 drift = n_pkt_fr − n_pkt_suc
33 throughput = n_pkt_suc × T_P/T_frame
```

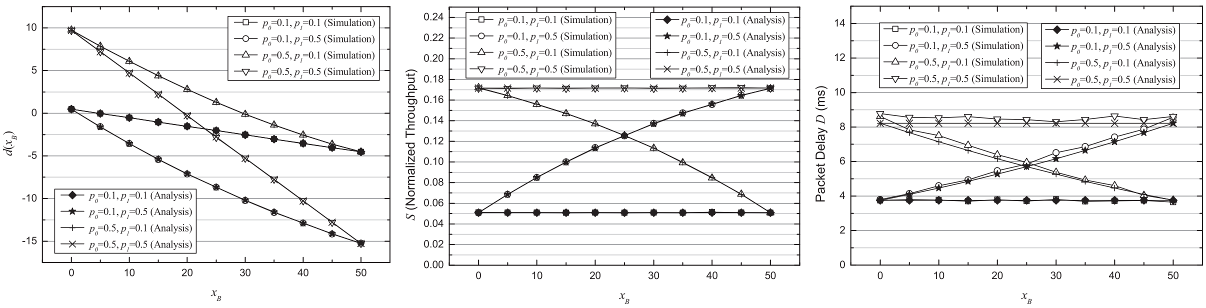
ets and the number of fresh packets, respectively. From the line 8 to 23, packets are generated and transmitted by both fresh nodes and backlogged nodes. The line from 9 to 15 are for the fresh nodes. In line 10, rand() function returns a single uniformly distributed random number between 0 and 1. If the return value is less or equal to p_0, the fresh node generates a packet and determines a slot for transmission by using getSlotIndex() function. This function performs time alignment mechanism described in Section 2.1 with the input value $t_{s,\,i}$ and returns slot index corresponding to the $t_{s,\,i}$. The return variable is assigned into *index* and slots[*index*] increases by one. Moreover, *n_pkt* and *n_pket_fr* also increase by one, respectively. The line from 16 to 22 are for the backlogged nodes. In line 17, if the return value of rand() is less or equal to p_1, the backlogged node generates a packet and determines a slot for transmission by using getRandNum() function, not getSlotIndex(). The getRandNum() function requires input value M and returns a single uniformly distributed random number between 1 to M. The return value is assigned into *index* and slot[*index*] and *n_pkt* increase by one. From line 25 to 30, the variable *n_pkt_suc* counts the number of successful packets. Finally, in line 32 and 33, *drift* and *throughput* are obtained as Eqs. (7) and (20), respectively.

Algorithm 3: Obtaining packet delay.

```
Input:N, p0, p1, T_frame, M, γ, T_P, x_B, t_s,i, t_s,v, τ_i,
slots[M]
n_retx = 0
while 1
    for i = 1; i <= M; i++
        slots[i] = 0;
    end for
    n_pkt = 0, n_pkt_fr = 0, n_pkt_back = 0
    for i = 1; i <= N − x_B − 1; i++
        if rand() <= p0
            n_pkt_fr++
        end if
    end for
    for i = 1; i <= x_B − 1; i++
        if rand() <= p1
            n_pkt_back++
        end if
    end for
    if n_retx == 0 // First transmission
        if rand() <= p1
            n_pkt_back++
        end if
        v_index = getSlotIndex(t_s,v)
    else // From the 2nd transmission
            if rand() <= p0
                n_pkt_fr++
            end if
        v_index = getRandNum(M)
    end if

    slots[v_index]++ // tx by virtual node
    n_pkt = n_pkt_fr + n_pkt_back
    for i = 1; i <= n_pkt;i++
        index = getRandNum(M)
        slots[index]++ // tx by other nodes
    end for
    if slots[v_index] == 1 // tx success
        if // first transmission is successful
            delay = γ + T_S(v_index − 1) − t_s,v + τ_i
        else
            delay = (T_frame − t_s,v)
                +T_frame(n_retx − 1)
                +γ + T_S(v_index − 1) + τ_i
        end if
        break // escape while loop
    else // retransmission is required
        n_retx++
    end if
end while
```

Second, Algorithm 3 shows the algorithm of the second simulator to obtain packet delay. To calculate the packet delay, we define a virtual node as one of the fresh nodes. The virtual node attempts to transmit a packet until success. Thus, the packet of virtual node is used to calcu-



(a) Drift (b) Throughput (c) Packet Delay

Fig. 7. Comparison between simulation and analysis (M=50, N=50).

late the packet delay. In line 2, *n_retx*, which denotes the number of retransmissions, is initialized by zero. Then the while loop starts and finished when transmission is successful. From line 4 to 7, slots[M] array, *n_pkt*, *n_pkt_fr*, and *n_pkt_back* are initialized. Then from line 8 to 17, $N - x_B - 1$ fresh nodes and $x_B - 1$ backlogged nodes, totally $N - 2$ nodes, generate their packet. Two nodes including a virtual node did not generate packets yet. Either a virtual node or the other node becomes a fresh node. The condition in line 18 means that a virtual node is a fresh node. Thus, the condition is satisfied, the backlogged node generates a packet through line 19 to 21 and the virtual node determines slot index, *v_index*, for transmission by using getSlotIndex() function. If the condition in line 18 is not satisfied, *i.e.*, the virtual nodes is a backlogged node and the other node is a fresh node, then the fresh node generates a packet through line 24 to 26 and the virtual node determines *v_index* for retransmission by using getRandNum() function. After generating packets from N nodes including a virtual node, transmission is performed through from line 30 to 35. In line 36, if the transmission of virtual node is successful, the condition is satisfied. Then the packet delay is calculated and stored in *delay* variable. If the transmission of the virtual node is successful at once, then the value of *delay* is obtained as in line 38. Otherwise, the *delay* is obtained considering the number of retransmissions, *n_retx*, as in lines from 40 to 42. After that, while loop is finished by break statement. However, the transmission of a virtual node is failed. Then *n_retx* increases by one and while loop restarts.

Fig. 7 presents the comparison between simulation and analysis when both M and N are 50. As shown, analytical results and simulation results of drift, throughput, and packet delay are almost same, when the same values of variables are used. Thus, our analytical models are verified.

5. Conclusions

In this paper, we simply designed BAS-ALOHA to support retransmission using B-ACK, and also derived analytical models for BAS-ALOHA for stability, throughput, and packet delay. Through time alignment with the estimated *PDC* and B-ACK, BAS-ALOHA can reduce guard time caused by large propagation delay, and can be used adequately for long propagation networks. All of our analytical models are fully mathematical, unlike existing models. Through numerical results, we showed the performance of BAS-ALOHA with various parameters: the number of slots in a frame, packet length, p_0, and p_1. We expect our analytical models to be a foundation for deriving fully mathematical models for variants of S-ALOHA using B-ACK.

Appendix A. Derivation of Eq. (40)

We use infinite geometrical progression in order to obtain equation $\sum_{i=2}^{\infty} E[D_i] \times P_S(i)$. To start the equation with $i = 1$, we rewrite the equation as

$$\begin{aligned}
&\sum_{i=2}^{\infty} E[D_i] \times P_S(i) \\
&= \sum_{i=2}^{\infty} \left\{ \left((i-1)T_{frame} + \frac{\gamma}{2} \right) (1-\rho)^{i-1} \rho \right\} \\
&= \sum_{i=2}^{\infty} \left\{ \left((i-1)T_{frame} + \frac{\gamma}{2} \right) (1-\rho)^{i-1} \rho \right\} + \frac{\gamma\rho}{2} - \frac{\gamma\rho}{2} \\
&= \sum_{i=1}^{\infty} \left\{ \left((i-1)T_{frame} + \frac{\gamma}{2} \right) (1-\rho)^{i-1} \rho \right\} - \frac{\gamma\rho}{2}
\end{aligned} \tag{A.1}$$

where $E[D_i]$ and $P_S(i)$ for $i \geq 2$ can be obtained by (39) and (22), respectively. The first term of (A.1) can be divided into two parts as

$$\begin{aligned}
&\sum_{i=1}^{\infty} \left\{ \left((i-1)T_{frame} + \frac{\gamma}{2} \right) \times (1-\rho)^{i-1} \rho \right\} \\
&= \sum_{i=1}^{\infty} (i-1)T_{frame} \times (1-\rho)^{i-1} \rho \\
&\quad + \sum_{i=1}^{\infty} \frac{\gamma(1-\rho)^{i-1}\rho}{2}.
\end{aligned} \tag{A.2}$$

In (A.2), the first term can be expressed as

$$\begin{aligned}
&\sum_{i=1}^{\infty} (i-1)T_{frame} \times (1-\rho)^{i-1} \rho \\
&= \sum_{i=1}^{\infty} iT_{frame}(1-\rho)^{i-1}\rho - \sum_{i=1}^{\infty} T_{frame}(1-\rho)^{i-1}\rho.
\end{aligned} \tag{A.3}$$

As shown in (A.3), this equation consists of two terms. The first term of (A.3) can be obtained by

$$\begin{aligned}\sum_{i=1}^{\infty} iT_{frame}(1-\rho)^{i-1}\rho &= \sum_{i=1}^{\infty} -\rho T_{frame}\frac{d}{d\rho}(1-\rho)^i \\ &= -\rho T_{frame}\sum_{i=1}^{\infty}\frac{d}{d\rho}(1-\rho)^i \\ &= -\rho T_{frame}\frac{d}{d\rho}\sum_{i=1}^{\infty}(1-\rho)^i \\ &= -\rho T_{frame}\frac{d}{d\rho}\left(\frac{1-\rho}{\rho}\right) \\ &= -\rho T_{frame}\frac{-1}{\rho^2} \\ &= \frac{T_{frame}}{\rho} \end{aligned} \quad \text{(A.4)}$$

and the second term of (A.3) can be obtained by

$$\begin{aligned}\sum_{i=1}^{\infty} T_{frame}(1-\rho)^{i-1}\rho &= \frac{\rho T_{frame}}{1-(1-\rho)} \\ &= T_{frame}. \end{aligned} \quad \text{(A.5)}$$

By combining (A.4) and (A.5), (A.3) can be rewritten as

$$\begin{aligned}\sum_{i=1}^{\infty}(i-1)T_{frame}(1-\rho)^{i-1}\rho &= \frac{T_{frame}}{\rho} - T_{frame} \\ &= T_{frame}\left(\frac{1}{\rho}-1\right). \end{aligned} \quad \text{(A.6)}$$

Using (A.6), we can rewrite (A.2) as

$$\begin{aligned}&\sum_{i=1}^{\infty}\left\{\left((i-1)T_{frame}+\frac{\gamma}{2}\right)\times(1-\rho)^{i-1}\rho\right\} \\ &= T_{frame}\left(\frac{1}{\rho}-1\right) + \sum_{i=1}^{\infty}\frac{\gamma(1-\rho)^{i-1}\rho}{2} \\ &= T_{frame}\left(\frac{1}{\rho}-1\right) + \frac{\gamma}{2}. \end{aligned} \quad \text{(A.7)}$$

Finally, by substituting (A.7) into (A.1), we obtain (A.8), which is expressed as

$$\begin{aligned}&\sum_{i=2}^{\infty} E[D_i]\times P_S(i) \\ &= T_{frame}\left(\frac{1}{\rho}-1\right) + \frac{\gamma}{2} - \frac{\gamma\rho}{2} \\ &= T_{frame}\left(\frac{1}{\rho}-1\right) + \frac{\gamma(1-\rho)}{2}. \end{aligned} \quad \text{(A.8)}$$

As shown in (A.8), the equation (40) is derived.

References

[1] H. Baek, J. Lim, S. Oh, Beacon-based slotted ALOHA for wireless networks with large propagation delay, IEEE Commun. Lett. 17 (11) (2013) 2196–2199.

[2] A. Syed, W. Ye, B. Krishnamachari, J. Heidemann, Understanding spatio-temporal uncertainty in medium access with ALOHA protocols, in: Proceedings of 2007 ACM WUWNet, 2007, pp. 41–48.

[3] L.F.M. Vieira, J. Kong, U. Lee, M. Gerla, Analysis of aloha protocols for underwater acoustic sensor networks, in: Proceedings of 2006 ACM WUWNet, 2006.

[4] D. Makrakis, K.M.S. Murthy, Spread slotted ALOHA techniques for mobile and personal satellite communication systems, IEEE J. Sel. Areas Commun. 10 (6) (1992) 985–1002.

[5] Y. Zhou, K. Chen, J. He, H. Guan, Enhanced slotted aloha protocols for underwater sensor networks with large propagation delay, in: Proceedings of 2011 IEEE Vehicular Technology Conference - Spring, 2011, pp. 1–5.

[6] P. Mandal, S. De, S.S. Chakraborty, A receiver synchronized slotted aloha for underwater wireless networks with imprecise propagation delay information, Ad Hoc Netw. 11 (4) (2013) 1443–1455.

[7] D. Pompili, I.F. Akyildiz, Overview of networking protocols for underwater wireless communications, IEEE Commun. Mag. 47 (1) (2009) 97–102.

[8] P. Casari, M. Zorzi, Protocol design issues in underwater acoustic networks, Comput. Commun. 34 (2) (2009) 170–180.

[9] L. Kleinrock, S.S. Lam, Packet switching in a multi access broadcast channel: performance evaluation, IEEE Trans. Commun. COM 23 (4) (1975) 410–422.

[10] R. Rom, M. Sidi, Multiple Access Protocols: Performance and Analysis, Springer-Verlag Inc., New York, 1990, pp. 49–53.

[11] R. Murali, B.L. Hughes, Random access with large propagation delay, IEEE/ACM Trans. Netw. 5 (6) (1997) 924–935.

[12] C. Kissling, On the stability of contention resolution diversity slotted ALOHA (CRDSA), in: Proceedings of 2011 IEEE Globecom, 2011, pp. 1–6.

[13] S.S. Lam, L. Kleinrock, Packet switching in a multi access broadcast channel: dynamic control procedures, IEEE Trans. Commun. COM 23 (9) (1975) 891–904.

[14] J.E. Wieselthier, A. Ephremides, L.A. Michaels, An exact analysis and performance evaluation of framed ALOHA with capture, IEEE Trans. Commun. 37 (2) (1989) 125–137.

[15] Northrop Grumman Corporation Information Technology Communication & Information Systems Division, Understanding Link-16: A Guidebook for New Users, NCTSI, San Diego, CA, (2001) pp. 5.47–5.49.

[16] L. Pu, Y. Luo, H. Mo, S. Le, Z. Peng, J.-H. Cui, Z. Jiang, Comparing underwater MAC protocols in real sea experiments, Comput. Commun. 56 (2015) 47–59.

Hoki Baek received the B.S. degree and the M.S. degree in Information and Computer Engineering from Ajou University, South Korea, in 2006 and 2008, respectively. He also received the Ph.D. degree in Computer Engineering from Ajou University in 2014. From March 2014 to February 2015, he served as a Post-Doctoral researcher in Jangwee Research Institute for National Defense of Ajou University. Since 2008, he has been participating in the Tactical Network Research Center of Ajou University. His research interests include Ad Hoc network, Tactical Data Link systems, military communication and networking technologies such as random access, distributed dynamic TDMA, MAC protocol, time synchronization, localization, and track fusion. Since March 2015, he has been with Ajou University as a Lecture Professor in the Department of Military Digital Convergence.

Jaesung Lim received the B.S. degree in electronic engineering from Ajou University, Korea, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), in 1985 and 1994, respectively. In 1985, he started as a Researcher at DAEWOO Telecommunication. In April 1988, he joined the Institute of DigiCom, and was engaged in research and development of data modem, radar signal processing and packet data systems. From 1995 to 1997, he served as a Senior Engineer in the Central Research and Development Center, SK Telecom, where he did research on wireless data communications for cellular and paging networks. Since March 1998, he has been with Ajou University, where he is a Professor with the Department of Computer Engineering, teaching and doing research in the areas of wireless, mobile, and tactical communications and networks. He has also been the Director of AJOU-TNRC (Tactical Networks Research Center) since 2006.