**Zheng, Feng; Kaiser, Thomas:**

**Adaptive Aloha anti-collision algorithms for RFID systems**

URN: urn:nbn:de:hbz:464-20161007-151942-1

PURL / DOI: http://dx.doi.org/10.1186/s13639-016-0029-7

WWW-Link: http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=42223



Quelle: EURASIP Journal on Embedded Systems, 2016:7; published: 12 April 2016

Zheng and Kaiser *EURASIP Journal on Embedded Systems* (2016) 2016:7
DOI 10.1186/s13639-016-0029-7

EURASIP Journal on Embedded Systems

RESEARCH Open Access

# Adaptive Aloha anti-collision algorithms for RFID systems

Feng Zheng* and Thomas Kaiser

**Abstract**

In this paper, we propose two adaptive frame size Aloha algorithms, namely adaptive frame size Aloha 1 (AFSA1) and adaptive frame size Aloha 2 (AFSA2), for solving radio frequency identification (RFID) multiple-tag anti-collision problem. In AFSA1 and AFSA2, the frame size in the next frame is adaptively changed according to the real-time collision rate measured in the current frame. It is shown that AFSA1 and AFSA2 can significantly improve the transmission efficiency of RFID systems compared to the static Aloha, and AFSA2 produces transmission efficiency similar to that of the electronic product code (EPC) Q-selection algorithm (Variant II), while the mean identification delay of AFSA2 is much smaller than that of EPC Q-selection algorithm (Variant II). It is also shown that the transmission efficiency of AFSA2 and EPC Variant II is very close to its upper bound which is obtained by assuming that the reader knows the number of unidentified tags. It is worth noting that when the threshold of the collision rate is chosen to be 0.5 or 0.6, AFSA2 can maintain the transmission efficiency well above 0.65 for the case of a typical EPC code length of 96 bits and for the investigated range of tag population, i.e., from 2 to 1000, while keeping the mean identification delay below ten transmit contentions. Very light computational burden at the reader is needed: the reader needs only to measure the collision rate in the current frame and then to double or halve the frame size accordingly. No additional computational burden is required at the tag side.

**Keywords:** RFID, Anti-collision, Aloha, Adaptive Aloha, Transmission efficiency, Mean identification delay

## 1 Introduction

With the development of the Internet of Things, radio frequency identification (RFID) has become a more and more active research field. In practical systems, we often meet the situation that there are many tags in the interrogation zone of a reader. If multiple tags simultaneously backscatter signals to the reader, a collision will occur. In principle, many advanced multiple channel accessing algorithms in wireless networks can be applied to RFID collision-resolution problem. However, passive RFID systems are highly asymmetric, i.e, the reader is resource rich, while tags have very limited storage and computing capabilities and are unable to hear the signal transmitted by other tags or to detect collisions. Therefore, channel access must be arbitrated by the reader [1–3]. Due to this fact, only basic anti-collision protocols, which can be broadly categorized into tree-splitting-based algorithms and Aloha-based algorithms, have been recommended in RFID protocols and implemented in practical RFID systems. Several extensive surveys about anti-collision algorithms for RFID systems can be found in [2–4].

This paper is focused on the study of Aloha-based anti-collision algorithms. In Aloha-based protocol, the whole interrogation period (maximal backoff time) is divided into $2^Q$ time slots, where $Q$ is an integer which is specified by the reader to tags through the reader-to-tag communication link. Each tag will independently, randomly, and uniformly select a time slot or backoff time, marked as an integer in the interval $[0, 2^Q - 1]$, after receiving an interrogation request from the reader. In the air interface protocol specified in [5], a tag will first backscatter its chosen integer, called tag handle, when the time reaches the backoff time which the tag selects. If only one tag handle is received by the reader, then the reader will send an "ACK" signal to inform that the tag can further backscatter its tag identity (ID) information. On the other hand, if two or more tag handles are received by the reader, a collision happens, the tag will be unacknowledged, and all

*Correspondence: feng.zheng@uni-due.de
Institute of Digital Signal Processing, University of Duisburg-Essen, 47057 Duisburg, Germany

the unacknowledged tags will independently select random backoff time again in the next round. This process is repeated until all the tags are finally identified and acknowledged by the reader [6–8].

As is well known, Aloha-based anti-collision algorithms work efficiently when the backoff size matches with the number of tags, but the performance of the algorithms becomes very poor when the number of tags changes in a wide range if the size of backoff time is fixed. Therefore, many dynamic Aloha anti-collision algorithms have been proposed to improve the system performance. The first dynamic frame-slotted Aloha protocol was proposed in [9] for general data networks. For RFID systems, different dynamic Aloha algorithms have been proposed in [10–15], where the frame size is dynamically adjusted according to the estimated number of tags. The difference in the aforementioned algorithms lies in that the tag number estimation methods are different. These approaches indeed yield better performance than the static Aloha protocol. However, they generally need many rounds of communications before the identification process to optimize the frame size [12].

Since estimating the number of tags plays a key role in the existing dynamic Aloha algorithms, several methods have been developed to estimate the number of tags. For example, in [14], a mean-square estimator was proposed to estimate tag number; in [16], a Bayesian approach was developed to estimate the number of tags, and in [11], a maximum likelihood estimator was used to estimate the number of tags based on partially observed frame contention data.

Consider a contention frame with $N$ tags and $L$ time slots. Each tag randomly, uniformly, and independently selects one of the $L$ time slots. It can be shown (see, e.g., [17]) that the expected number of colliding transmission slots, denoted by $\mu_c(N,L)$, is given by

$$\mu_c(N,L) = L - L\left(1-\frac{1}{L}\right)^N - N\left(1-\frac{1}{L}\right)^{N-1}. \quad (1)$$

From Eq. (1), it can be easily seen that the number of tags can be estimated, in principle, from the *expected* number of colliding transmissions. However, to measure the *expected* number of colliding transmissions, it needs many rounds of communications, which is resource consuming.

In this paper, we propose two algorithms to improve the performance of dynamic Aloha anti-collision algorithms for RFID systems. In our approach, the frame size is adjusted according to the collision rate at the current layer (the concept layer will be defined in the sequel) instead of the estimated number of tags. Therefore, the estimator for predicting the number of tags is not required, while the collision rate can be easily measured at the reader side. Since the frame size is adjusted through changing the value of $Q$ parameter, the frame size is doubled or halved according to the current status of the collision rate.

Generally, two performance metrics, i.e., mean identification delay (MID) and transmission efficiency, are used to characterize the performance of an anti-collision algorithm [18]. The MID, denoted with $\bar{d}_{\mathrm{tr}}$, describes the behavior that after how many transmit contentions in average a tag's ID can be successfully delivered to the reader. Here, a *transmit contention* (TC) means that the reader sends the command Query or QueryAdjust *once* so that all the uninventoried tags will select new competing random time slots. This performance metric is useful when the durations needed for one successful, colliding, or idle transmission are roughly of the same value. Let $T_s$, $T_c$, and $T_i$ denote, respectively, the time needed for *one* successful, colliding, or idle transmission, and let $N_{\mathrm{suc}}$, $N_{\mathrm{col}}$, and $N_{\mathrm{idle}}$ denote, respectively, the number of successful, colliding, or idle transmissions when a given number of tags are completely inventoried by an anti-collision algorithm. Let the number of tags be $N$. Then, the transmission efficiency of this Aloha anti-collision algorithm is defined as

$$E_{\mathrm{Aloha}}(N) = \frac{N_{\mathrm{suc}}T_s}{N_{\mathrm{suc}}T_s + N_{\mathrm{col}}T_c + N_{\mathrm{idle}}T_i} \quad (2)$$

$$= \frac{1}{1+\frac{N_{\mathrm{col}}}{N}F_c + \frac{N_{\mathrm{idle}}}{N}F_i}, \quad (3)$$

where we have used the fact that $N_{\mathrm{suc}} = N$, and $F_i$ and $F_c$ are the fractional times of $T_i$ and $T_c$, respectively, compared with $T_s$, i.e.,

$$F_i = \frac{T_i}{T_s}, \quad F_c = \frac{T_c}{T_s}.$$

Generally, $F_i$ and $F_c$ are much smaller than unity in Aloha-based anti-collision algorithms if the specifications in [5] are adopted.

The rest of the paper is organized as follows. In Section 2, RFID link timing is briefly reviewed, which will provide an illustration for the ranges of the values of $F_i$ and $F_c$. In Section 3, the performance of static framed Aloha anti-collision algorithm is investigated. In Section 4, two adaptive framed Aloha algorithms are proposed, and their performance is presented and compared to both electronic product code (EPC) Q-selection algorithm and an ideal system. Finally, concluding remarks are drawn in Section 5.

## 2 RFID link timing

To show the relationship among the parameters $T_s$, $T_c$, and $T_i$, let us have a look on the RFID link timing as shown in the EPC protocol [5]. Figure 1 illustrates the time used for three kinds of reader tag communications: successful transmission, collision, and idle transmission. As seen from Fig. 1, a tag will first backscatter its handle

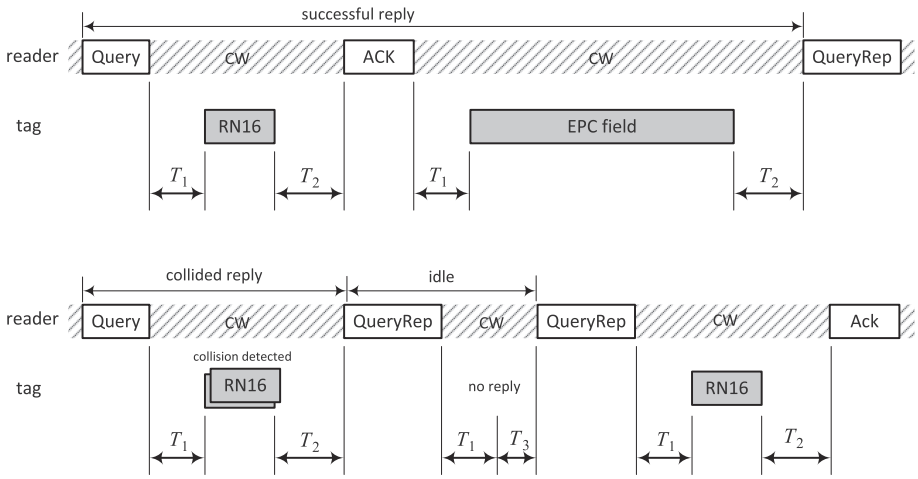


**Fig. 1** An illustration for RFID link timing, where parameters $T_1$, $T_2$, and $T_3$ denote relevant signal transmission/processing time and waiting time, respectively, CW stands for continuous wave, and the EPC field contains codes storedCRC (16 bits), storedPC (16 bits), EPC, and optional XPC word or words. The command "Query" can be also the command "QueryAdjust"

to the reader after receiving the reader's "Query" command. If the reader receives only one tag handle, it will send an acknowledgement signal ACK to the tag. After receiving this ACK signal, the tag will then backscatter its tag ID information. This process constitutes a successful transmission. If the reader receives two or more tag handles, a collision happens, and the collided tags will wait for new reader's command Query or "QueryAdjust" to select a new backoff time for the next round of inventory. If the reader receives no signal after a given waiting period, it will re-send the commands "QueryRep," Query, or QueryAdjust. This constitutes an idle transmission.

From Fig. 1, we can see that

$$T_{\mathrm{i}} = T_{\mathrm{Qx}} + T_1 + T_3, \tag{4}$$

$$T_{\mathrm{c}} = T_{\mathrm{Qx}} + T_1 + T_2 + T_{\mathrm{RN16}}, \tag{5}$$

$$T_{\mathrm{s}} = T_{\mathrm{Qx}} + T_1 + T_2 + T_{\mathrm{RN16}} + T_{\mathrm{ACK}} + T_1 + T_2 + T_{\mathrm{EPC}}, \tag{6}$$

where $T_{\mathrm{RN16}}$, $T_{\mathrm{ACK}}$, and $T_{\mathrm{EPC}}$ denote the time needed for the transmission of the commands/codes RN16, ACK, and EPC field, respectively, and $T_{\mathrm{Qx}}$ stands for

$$T_{\mathrm{Qx}} = \begin{cases} T_{\mathrm{Query}} & \text{when a successful/collided/idle reply happens after the reader's command Query,} \\ T_{\mathrm{QueryAdjust}} & \text{when a successful/collided/idle reply happens after the reader's command QueryAdjust,} \\ T_{\mathrm{QueryRep}} & \text{when a successful/collided/idle reply happens after the reader's command QueryRep,} \end{cases} \tag{7}$$

with $T_{\mathrm{Query}}$, $T_{\mathrm{QueryAdjust}}$, and $T_{\mathrm{QueryRep}}$ being the times needed for the transmission of the commands Query, QueryAdjust, and QueryRep, respectively.

The durations $T_1$, $T_2$, and $T_3$ are related with the parameters in the reader-to-tag physical interface. Let $T_{S_0}$ and $T_{S_1}$ denote the duration of data symbol "0" and the duration of data symbol "1," respectively. The ranges for $T_{S_0}$ and $T_{S_1}$ are as follows [5]:

$$6.25\ \mu\mathrm{s} \le T_{S_0} \le 25\ \mu\mathrm{s},$$
$$1.5T_{S_0} \le T_{S_1} \le 2T_{S_0}.$$

A reader starts the signaling for the reader-to-tag link with either a preamble or a frame sync. The preamble consists of four parts: a delimiter, data 0, reader-to-tag (R→T) calibration with duration $T_{\mathrm{RTcal}}$, and tag-to-reader (T→R) calibration with duration $T_{\mathrm{TRcal}}$. These parameters are specified as follows [5]:

$$2.5T_{S_0} \le T_{\mathrm{RTcal}} = T_{S_0} + T_{S_1} \le 3T_{S_0},$$
$$1.1T_{\mathrm{RTcal}} \le T_{\mathrm{TRcal}} \le 3T_{\mathrm{RTcal}}.$$

The T→R calibration, together with the command *divide ratio* (DR), specifies a tag's backscatter link frequency (BLF) or backscatter-link pulse-repetition frequency. The tag measures $T_{\mathrm{TRcal}}$ and then computes BLF as [5]

$$\mathrm{BLF} = \frac{\mathrm{DR}}{T_{\mathrm{TRcal}}}.$$

The backscatter-link pulse-repetition interval, denoted as $T_{\mathrm{pri}}$, is calculated from

$$T_{\mathrm{pri}} = 1/\mathrm{BLF} = \frac{T_{\mathrm{TRcal}}}{\mathrm{DR}}.$$

There are two choices for the value of DR: DR = 8 or DR = 64/3.

The data rate of the backscatter link depends on the modulation type. For simpleness, we assume that the FM0 baseband modulation is used throughout this paper. Hence, the number of the subcarrier cycles per symbol is 1, and the data rate equals BLF.

From Table 6.28 of [5], it is seen that the commands Query, QueryRep, QueryAdjust, and ACK are of length of 22, 4, 9, and 18 bits, respectively. Therefore, the times needed for the transmission of these commands are

$$T_{\text{Query}} = 22T_{\text{pri}}, \tag{8}$$
$$T_{\text{QueryAdjust}} = 9T_{\text{pri}}, \tag{9}$$
$$T_{\text{QueryRep}} = 4T_{\text{pri}}, \tag{10}$$
$$T_{\text{ACK}} = 18T_{\text{pri}}. \tag{11}$$

According to Annex L of [5], the size of the EPC memory is defined by tag manufacturers. The minimum size is 32 bits, to contain a 16-bit StoredCRC and a 16-bit StoredPC. Further EPC memory is generally provided to contain an EPC whose length ranges from 16 to 496 bits (if a tag does not support XPC functionality) or to 464 bits (if a tag supports XPC functionality), as well as an optional XPC word or words. Therefore, the time needed for the transmission of EPC memory is

$$T_{\text{EPC}} = (32 + x_{\text{EPC}})T_{\text{pri}}, \tag{12}$$

where $x_{\text{EPC}}$ denotes the length (in bits) of the variable part of the EPC code.

From Table 6.16 of [5], it is seen that

$$T_1 = \max\left\{T_{\text{RTcal}}, 10T_{\text{pri}}\right\}, \tag{13}$$
$$3T_{\text{pri}} \le T_2 \le 20T_{\text{pri}}, \tag{14}$$

while $T_3$ depends on the design of a reader.

Substituting Eqs. (11) and (12) into Eqs. (5) and (6), we obtain the following:

$$T_{\text{c}} = T_{\text{Qx}} + T_1 + T_2 + 16T_{\text{pri}}, \tag{15}$$
$$T_{\text{s}} = T_{\text{Qx}} + 2(T_1 + T_2) + (66 + x_{\text{EPC}})T_{\text{pri}}. \tag{16}$$

Substituting Eqs. (7)–(10), (13), and (14) into Eqs. (4), (15), and (16), we can calculate the values of parameters $F_{\text{i}}$ and $F_{\text{c}}$.

Figure 2 shows the ranges of $F_{\text{i}}$ and $F_{\text{c}}$, where $T_{S_0} = 20\,\mu\text{s}$, $T_3 = 20T_{\text{pri}}$, DR = 8, $T_{S_1} = 1.75T_{S_0}$, and $T_{\text{TRcal}} = 2T_{\text{RTcal}}$. Changing the values of these parameters in the specified ranges, we can also observe that $F_{\text{i}}$ and $F_{\text{c}}$ are in the similar ranges as shown in Fig. 2.

From Fig. 2, we can see that $T_{\text{i}}$ and $T_{\text{c}}$ are only small fractions of $T_{\text{s}}$ when a typical length of 96 bits is used for EPC code.

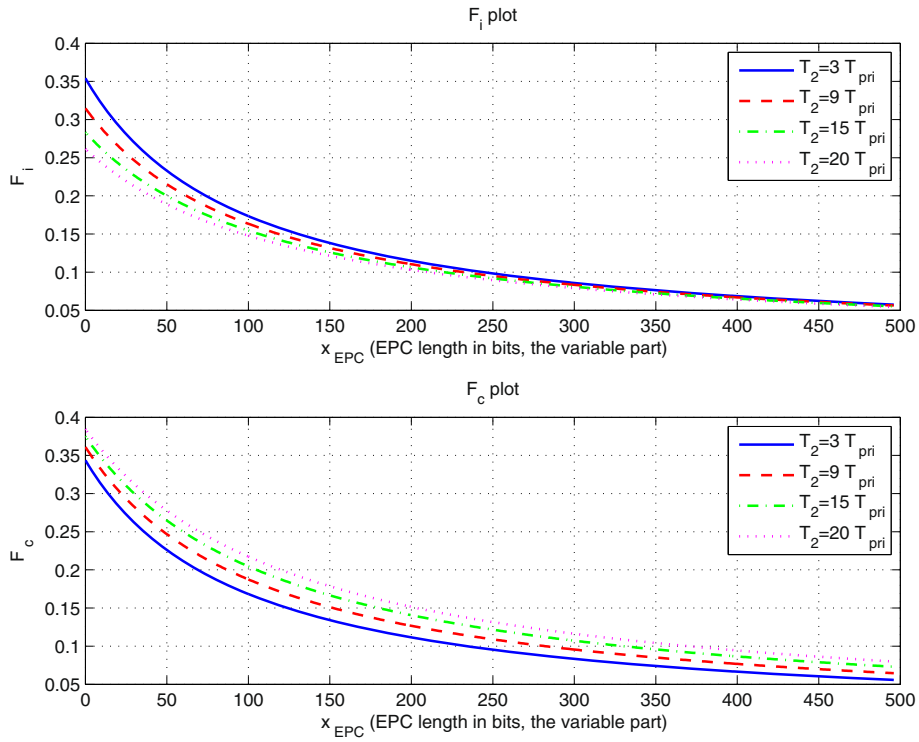


**Fig. 2** An illustration for the ranges of $F_{\text{i}}$ and $F_{\text{c}}$, where $T_{\text{QueryRep}}$ is used for $T_{Qx}$

## 3 Static-framed Aloha algorithm for RFID systems

In this section, the performance of the static-framed Aloha anti-collision algorithm for RFID systems will be investigated. In a framed Aloha, one frame consists of several, say $L$, time slots, and a tag randomly chooses a time slot among these $L$ time slots to transmit. Once a collision happens, it will again randomly choose a time slot and wait in the next frame (which is also called next layer in the sequel) to transmit. In a framed Aloha algorithm, we say that the tags are located in the same *layer* if they have experienced the same number of contentions to transmit.

A by-product of the framed Aloha is that it is easy for the reader to calculate the number of colliding slots, which will enable many adaptive random access protocols.

In a framed Aloha scheme, the frame length may be either fixed or variable depending on the particular system implementation. The former is called static-framed Aloha, while the latter is called dynamic framed Aloha.

Theoretically, the idle transmission probability and collision probability for a given number of tags can be derived based on classical probability theory. Then, the transmission efficiency can be calculated based on the idle transmission probability and collision probability. For details, readers are referred to reference [18]. However, the analytical results can be used to illustrate the system performance only when the number of tags is small due to the fact that the involved binomial coefficient can be calculated only up to a limited range of $N$ in Matlab. Therefore, we resort to simulations to illustrate the system performance.

Figure 3 illustrates the transmission efficiency $E_{\text{Aloha}}$ and MID $\bar{d}_{\text{tr}}$ of the static Aloha versus the number of tags for different frame size $L$, where $T_{S_0} = 20\,\mu$s, DR$=$ 8, $T_{S_1} = 1.75T_{S_0}$, $T_2 = T_3 = 20T_{\text{pri}}$, and $T_{\text{TRcal}} = 2T_{\text{RTcal}}$.

Figure 3a shows that only when the number of tags $N$ falls into a very limited range can the static Aloha algorithm achieve a reasonably good transmission efficiency. For example, if the transmission efficiency is required to be greater than 0.5, $N$ should be in the range as shown in Table 1 for different $L$.

When $N$ is out of the above range, the transmission efficiency decreases rapidly to some very low values. This is not satisfactory in practical RFID applications.

Figure 3b shows that the MID $\bar{d}_{\text{tr}}$ of the static Aloha increases with the number of tags exponentially. For example, for the case of $L = 32$, to inventory $N = 370$ tags, the reader needs to transmit more than $10^4$ times of Query or QueryAdjust command in average. This is also unsatisfactory.

## 4 Adaptive frame size Aloha algorithms

It is observed from Fig. 3 that when the number of time slots in a frame matches properly with the number of tags in the interrogation zone, the transmission efficiency approaches to its maximal value. Based on this idea, a dynamic frame-slotted Aloha protocol was first proposed about three decades ago for general data networks and has been applied to RFID systems in recent years. In dynamic frame-slotted Aloha protocols, it is generally required to estimate accurately the number of tags to be inventoried, which is not easy. In this section, we propose two algorithms, called adaptive frame size Aloha 1 (AFSA1) and adaptive frame size Aloha 2 (AFSA2), to give another solution for the dynamic Aloha anti-collision problem.

AFSA1 and AFSA2 work in the following way. Suppose that the frame size at the current layer is $L$. Define the collision rate $R_{\text{c}}$ at the current layer as

$$R_{\text{c}} = \frac{N_{\text{c}}}{L}, \tag{17}$$

where $N_{\text{c}}$ is the number of colliding transmissions in the *current* layer. The frame size at next layer will be adaptively changed according to the real-time measured collision rate $R_{\text{c}}$. To start AFSA1 or AFSA2, three parameters, namely $L_0$, $L_{\max}$, and $R_{\text{c0}}$, need to be specified, where $L_0$ is the frame size at the starting frame, $L_{\max}$ is a predefined maximal frame size, and $R_{\text{c0}}$ is a threshold for $R_{\text{c}}$. Generally, $L_0$ and $L_{\max}$ are set to be some powers of 2.

**Algorithm 1 AFSA1**

**Step 0** Initial setting: given numbers $L_0$, $L_{\max}$, and $R_{\text{c0}}$. Set $L = L_0$.

**Step 1** Check frame status: at every frame, the reader calculates the collision rate $R_{\text{c}}$ according to Eq. (17).

**Step 2** Adjust the frame size according to the following law:

$$L := \begin{cases} 2L & \text{if } R_{\text{c}} > R_{\text{c0}} \text{ and } L < L_{\max}, \\ L & \text{otherwise, i.e., } L \text{ is kept unchanged.} \end{cases}$$

Go to next layer.

**Algorithm 2 AFSA2**

**Step 0** Initial setting: given numbers $L_0$, $L_{\max}$, and $R_{\text{c0}}$. Set $L = L_0$.

**Step 1** Check frame status: at every frame, the reader calculates the collision rate $R_{\text{c}}$ according to Eq. (17).

**Step 2** Adjust the frame size according to the following law:

$$L := \begin{cases} 2L & \text{if } R_{\text{c}} > R_{\text{c0}} \text{ and } L < L_{\max}, \\ L/2 & \text{if } R_{\text{c}} < R_{\text{c0}}/2 \text{ and } L > L_0, \\ L & \text{otherwise, i.e., } L \text{ is kept unchanged.} \end{cases}$$

Go to next layer.

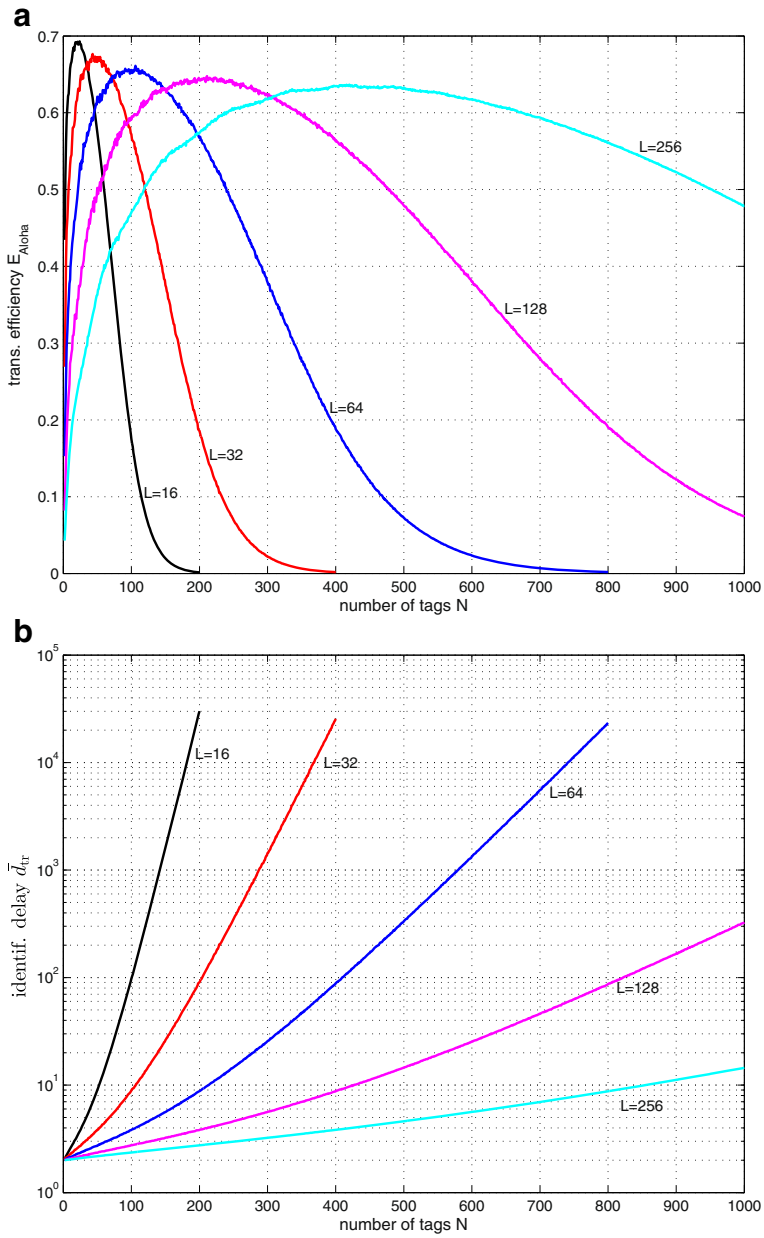


**Fig. 3 a**, **b** The transmission efficiency $E_{\text{Aloha}}$ and MID $\bar{d}_{\text{tr}}$ (in TC) of the static Aloha vs. the number of tags for different frame size, where $x_{\text{EPC}} = 64$, which leads to $F_{\text{i}} = 0.1753$ and $F_{\text{c}} = 0.2577$

**Table 1** The ranges of $N$ in which the transmission efficiency of the static-framed Aloha is greater than 0.5 versus different frame sizes $L$

| Frame size $L$ | Range of $N$ (for Fig. 3, $x_{\text{EPC}} = 64$) |
|---|---|
| 16 | [3, 59] |
| 32 | [8, 119] |
| 64 | [22, 239] |
| 128 | [52, 479] |
| 256 | [119, 952] |

The difference between AFSA1 and AFSA2 lies in that the frame size will not be reduced in AFSA1 once it is increased to $L_{\max}$, while in AFSA2, the frame size can be increased or decreased between $L_0$ and $L_{\max}$ depending on the collision rate. Therefore, the number of idle slots is also well controlled in AFSA2 if the parameter $R_{\text{c0}}$ is selected properly.

The exponential frame size adjustment method in AFSA1 and AFSA2 is borrowed from the idea of distributed coordination function in CSMA/CA algorithm used in IEEE 802.11 [19]. Another reason for this choice is

that it can be easily implemented in RFID by adjusting the $Q$ parameter in RFID protocol [5].

Note that the overhead to implement AFSA1 or AFSA2 in RFID is negligible: the reader needs only to measure the collision rate in the current frame and then to double or halve the frame size accordingly. No additional computational burden is needed at the tag.

To compare the performance of the proposed algorithms and the anti-collision algorithm recommended in the EPC protocol [5], we summarize the latter as follows. In the the EPC protocol [5, Annex D], the frame size is set to be $L = 2^Q$, where $Q$ is an integer and can change from 4 to 15 inclusive according to another real number $Q_{\text{fp}}$, i.e., $Q = \texttt{round}(Q_{\text{fp}})$ with round standing for the function of being rounded to the nearest integer. The number $Q_{\text{fp}}$ is dynamically changed in terms of the transmission status at every time slot according to the following law:

$$Q_{\text{fp}} := \begin{cases} Q_{\text{fp}} & \text{if the transmission at the time slot is successful,} \\ \max(0, Q_{\text{fp}} - \Delta) & \text{if the transmission at the time slot is idle,} \\ \min(15, Q_{\text{fp}} + \Delta) & \text{if the transmission at the time slot is colliding,} \end{cases} \tag{18}$$

where $\Delta$ is a pre-specified parameter. The initial value for $Q_{\text{fp}}$ is set to be $Q_{\text{fp}} = 4.0$. Typical values of $\Delta$ is $0.1 < \Delta < 0.5$, and $\Delta$ can be set to be large (or small) when $Q$ is small (or large). In our simulations, we choose two kinds of $\Delta$. The first choice is that $\Delta$ is a constant: $\Delta = 0.3$. The second choice is that $\Delta$ itself is dynamically changed according to the following:

$$\Delta = \begin{cases} 0.45 \text{ if } Q \leq 10, \\ 0.15 \text{ if } Q \geq 11. \end{cases} \tag{19}$$

The change for the values of $Q$ is implemented through the reader's command QueryAdjust. It is not specified in the protocol when to send the command QueryAdjust. We simulate two cases: (i) The command QueryAdjust is sent out after finishing all the transmit contentions in one layer. This case is referred to as EPC Variant I. (ii) The command QueryAdjust is sent out after the reader's receiving the tags' responses at every time slot. This case is referred to as EPC Variant II. It will be seen that the time instant of sending the command QueryAdjust affects the system performance greatly.

To provide another comparison, we study the performance of an ideal system where it is assumed that the reader has perfect knowledge about the number of unidentified tags. This case should provide an upper bound for the transmission efficiency of relevant systems. In this case, it can be easily shown (see, e.g., [18]) that the optimal frame size, denoted by $L_{\text{opt}}$, is given by

$$L_{\text{opt}} = N_{\text{unid}}, \tag{20}$$

where $N_{\text{unid}}$ stands for the number of unidentified tags after the transmit contentions of a layer. Considering the implementation mechanism of adjusting the frame size in EPC protocol, $L_{\text{opt}}$ should be some power of 2. Therefore, we can set $L_{\text{opt}}$ as follows:

$$L_{\text{opt}} = 2^{\texttt{round}(\log_2 N_{\text{unid}})}. \tag{21}$$

In this paper, both Schemes (20) and (21) will be simulated. For convenience, the Scheme (20) is denoted by "Perfect I," while the scheme (21) "Perfect II."

In all the simulation results, we choose $T_{S_0} = 20\,\mu$s, DR$= 8$, $T_{S_1} = 1.75T_{S_0}$, $T_{\text{TRcal}} = 2T_{\text{RTcal}}$, $T_2 = T_3 = 20T_{\text{pri}}$, $L_0 = 2^4 = 16$, and $L_{\max} = 2^{15} = 32,768$. This setup for the above parameters is either specified by the EPC protocol, such as $L_0$ and $L_{\max}$, or chosen in the corresponding ranges as specified by the EPC protocol, such as $T_{S_0}$, DR, $T_{S_1}$, $T_{\text{TRcal}}$, and $T_2$. Our other simulations, not included in this paper, show that for this latter group of free-chosen parameters, the system produces performance similar to that as being presented here if we choose other values for them in the specified ranges.

Figures 4, 5, and 6 illustrate the transmission efficiency of AFSA1 and AFSA2 for three representative lengths of $x_{\text{EPC}}$: $x_{\text{EPC}} = 0$, 64, and 496, respectively. The case of $x_{\text{EPC}} = 0$ stands for the extreme short EPC load, which leads to comparatively large $F_{\text{i}}$ and $F_{\text{c}}$: $F_{\text{i}} = 0.2615$ and $F_{\text{c}} = 0.3846$, for the chosen typical values of other reader-to-tag link parameters. The case of $x_{\text{EPC}} = 496$ stands for the extreme long EPC load, which leads to comparatively small $F_{\text{i}}$ and $F_{\text{c}}$: $F_{\text{i}} = 0.0543$ and $F_{\text{c}} = 0.0799$. The case of $x_{\text{EPC}} = 64$ stands for a typical EPC load, which leads to a typical $F_{\text{i}}$ and $F_{\text{c}}$: $F_{\text{i}} = 0.1753$ and $F_{\text{c}} = 0.2577$.

From Figs. 4, 5, and 6, we can see the following phenomena.

- Both AFSA1 and AFSA2 provide much better performance than the static Aloha algorithm. Actually, the transmission efficiency of AFSA1 and AFSA2 is well above 0.5 for all $N$ in the investigated range, i.e., from 2 up to 1000, when $x_{\text{EPC}} = 64$.
- When $N$ is very small, say $N \leq 40$, AFSA1 and AFSA2 produce almost the same performance, but when $N$ is large, i.e., $N > 80$, AFSA2 yields much better performance than AFSA1.
- By choosing $R_{\text{c0}}$ to be 0.5 or 0.6, AFSA2 gives stable and high transmission efficiency, while choosing $R_{\text{c0}}$ to be low or high, e.g., 0.4, 0.8, or 0.9, AFSA2 gives oscillating (with $N$) and low transmission efficiency.
- AFSA2 works well for a large range of $N$. For example, when $80 \leq N \leq 1000$, and choosing $R_{\text{c0}}$ to be 0.5 or 0.6, the transmission efficiency of the system is well above 0.55, 0.65, and 0.85 when $x_{\text{EPC}} = 0$, 64, and 496, respectively.

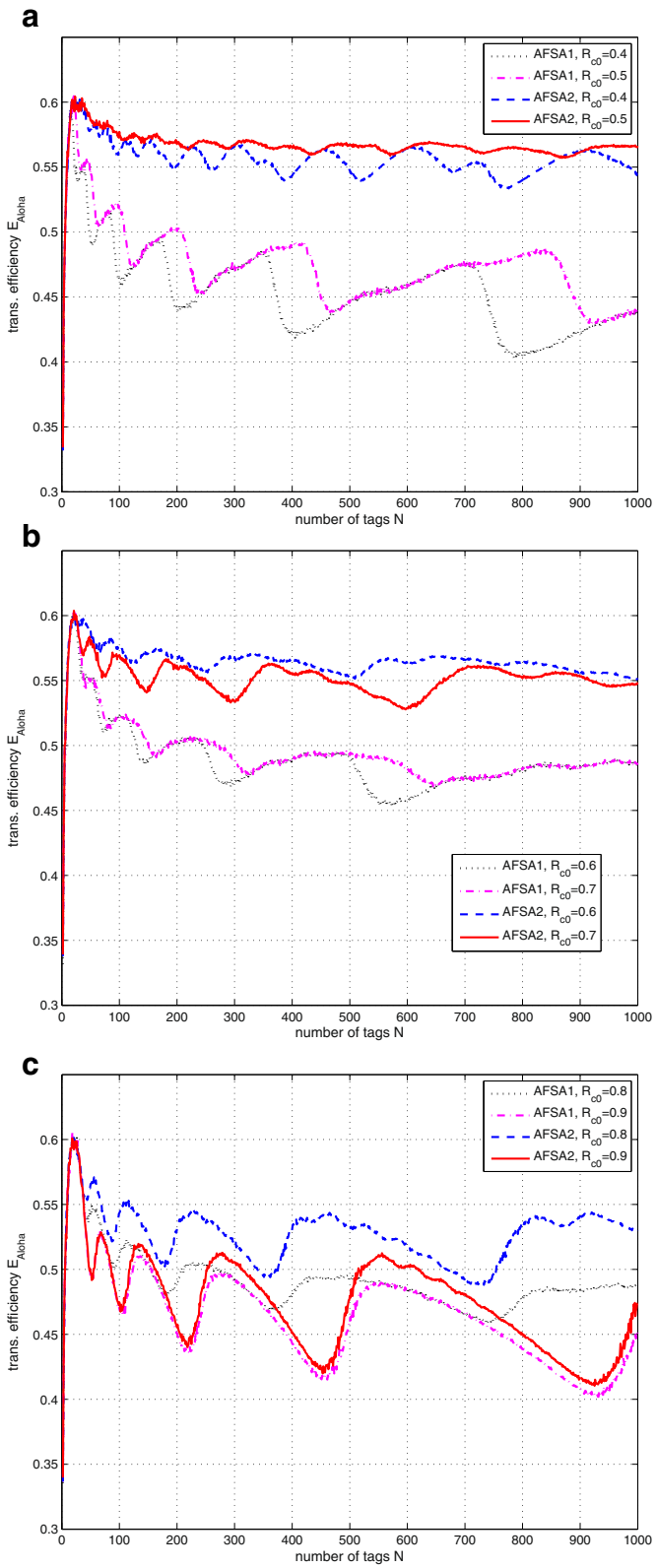


**Fig. 4 a–c** The transmission efficiency $E_{\text{Aloha}}$ of AFSA1 and AFSA2 vs. the number of tags for different $R_{c0}$, where $x_{\text{EPC}} = 0$

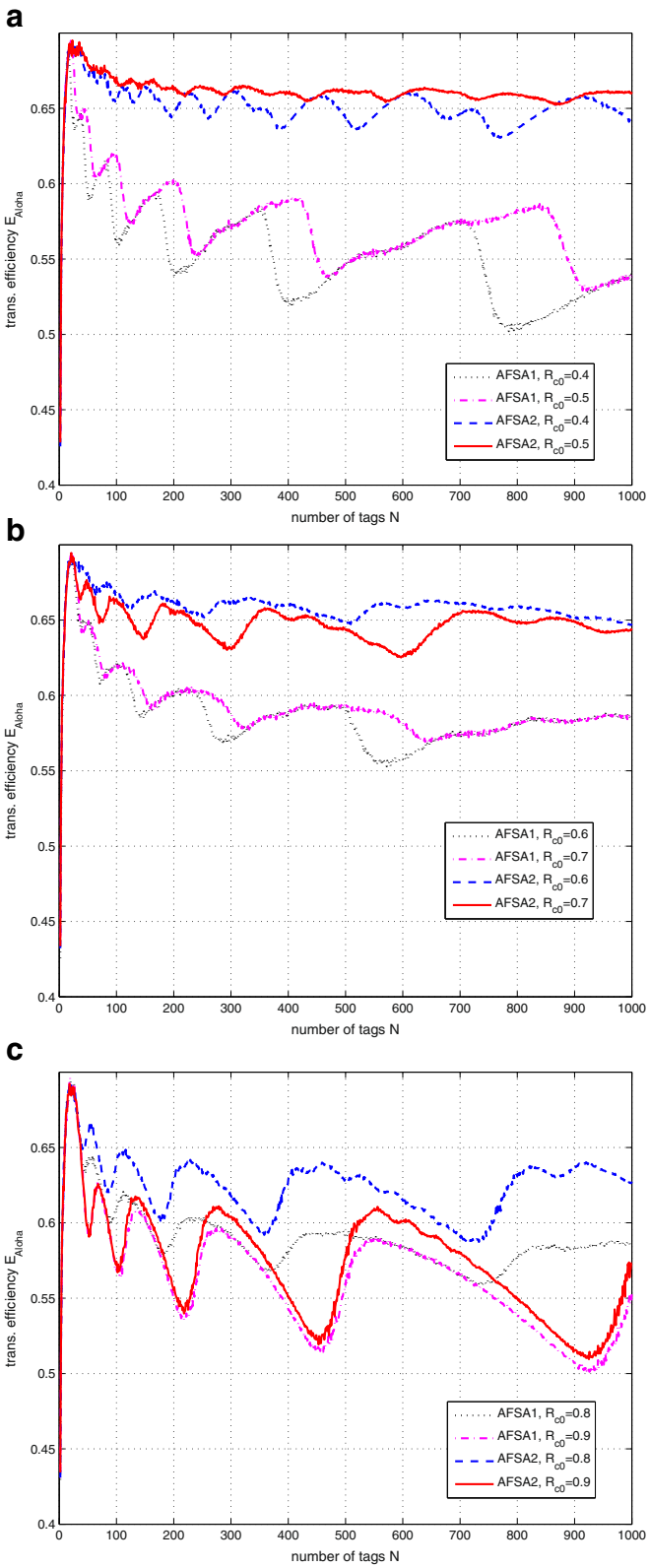


**Fig. 5 a–c** The transmission efficiency $E_{Aloha}$ of AFSA1 and AFSA2 vs. the number of tags for different $R_{c0}$, where $x_{EPC} = 64$

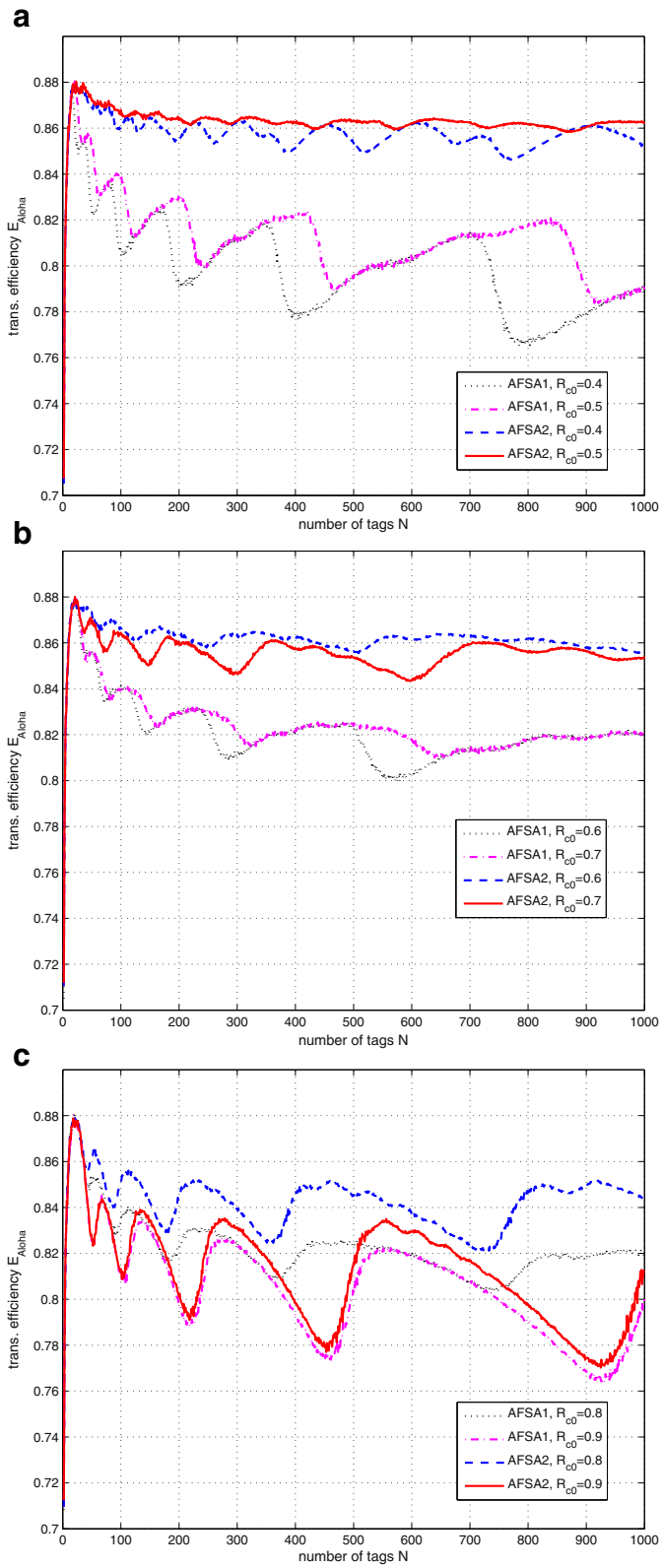


**Fig. 6 a–c** The transmission efficiency $E_{\text{Aloha}}$ of AFSA1 and AFSA2 vs. the number of tags for different $R_{c0}$, where $x_{\text{EPC}} = 496$

- The transmission efficiency of AFSA1 oscillates with $N$. This is because the mechanism of reducing the frame size is not implemented in AFSA1. Suppose that a series (for different layers) of local optimal frame sizes has been found by AFSA1 for some $N = N_0$ at which $E_{\text{Aloha}}$ achieves its local maximal value. Then, when $N = N_0 + 1$, the same series of frame sizes will be likely produced by AFSA1. However, this series of frame sizes will not be optimal for $N = N_0 + 1$ due to higher collision rate than the collision rate for the case $N = N_0$. This process continues until $N$ reaches another value, say $N = N_1$, at which $E_{\text{Aloha}}$ achieves its local minimal value.

Figures 7 and 8 show comparisons between AFSA2 and EPC Q-selection algorithm for $x_{\text{EPC}} = 64$ as a representative case. For AFSA2, we choose $R_{c0} = 0.5$. For the EPC Q-selection algorithm, two aforementioned variants, i.e., EPC Variant I and EPC Variant II, and two possible choices for the value of $\Delta$ are showed. For the changing

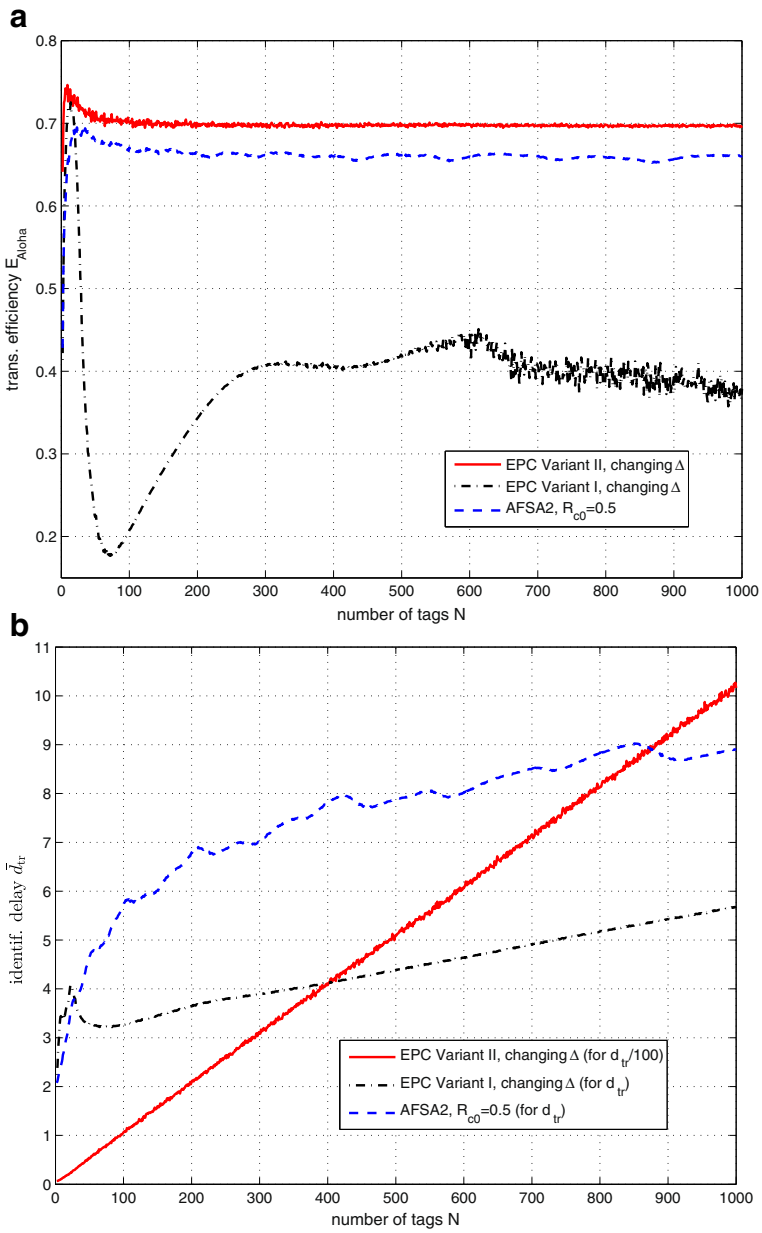


**Fig. 7** A comparison between AFSA2 and EPC Q-selection algorithm, where $R_{c0} = 0.5$, $x_{\text{EPC}} = 64$, and $\Delta$ changes according to the values of $Q$. **a** Transmission efficiency $E_{\text{Aloha}}$. **b** Mean identification delay $\bar{d}_{\text{tr}}$ (in TC). Notice that in this figure, the value of $\bar{d}_{\text{tr}}/100$, instead of $\bar{d}_{\text{tr}}$ itself, is plotted for EPC Variant II

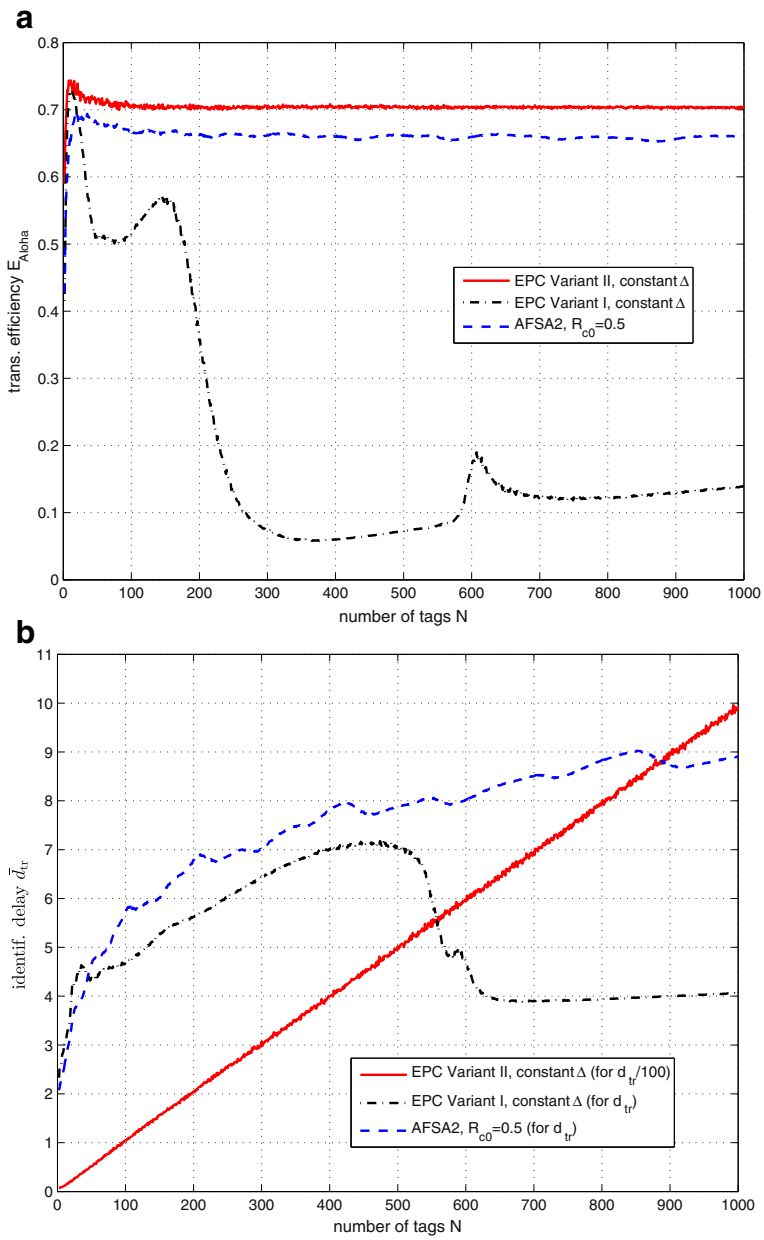


**Fig. 8** A comparison between AFSA2 and EPC Q-selection algorithm, where $R_{c0} = 0.5$, $x_{EPC} = 64$, and $\Delta = 0.3$ is a constant. **a** Transmission efficiency $E_{Aloha}$. **b** Mean identification delay $\bar{d}_{tr}$ (in TC). Notice that in this figure, the value of $\bar{d}_{tr}/100$, instead of $\bar{d}_{tr}$ itself, is plotted for EPC Variant II

$\Delta$ (in Fig. 7), $\Delta$ is changed according to the law given by Eq. (19). For the constant $\Delta$ (in Fig. 8), $\Delta$ is fixed to be 0.3. From Figs. 7a and 8a, we can see that the transmission efficiency (around 0.7 in the stable value, i.e., for large $N$) of EPC Variant II is a little higher than that of AFSA2 (around 0.66 in the stable value), while EPC Variant I performs very poor (its transmission efficiency is around 0.4 at large $N$ for the case of changing $\Delta$ and below 0.2 at large $N$ for the case of constant $\Delta$). From Figs. 7b and 8b, we can see that the MID $\bar{d}_{tr}$ of AFSA2 is below ten TCs when $N \leq 1000$, while $\bar{d}_{tr}$ of EPC Variant II increases linearly with $N$, and when $N = 1000$, the MID is around 1000 TCs. This means that, for EPC Variant II, the reader spends too much time on the transmission of QueryAdjust command, which is resource consuming for tags, since after receiving every QueryAdjust command, the tag needs to reselect a random time slot as its backoff time.

Figure 9 shows that the transmission efficiency and MID of Perfect I and Perfect II. It can be seen that for the

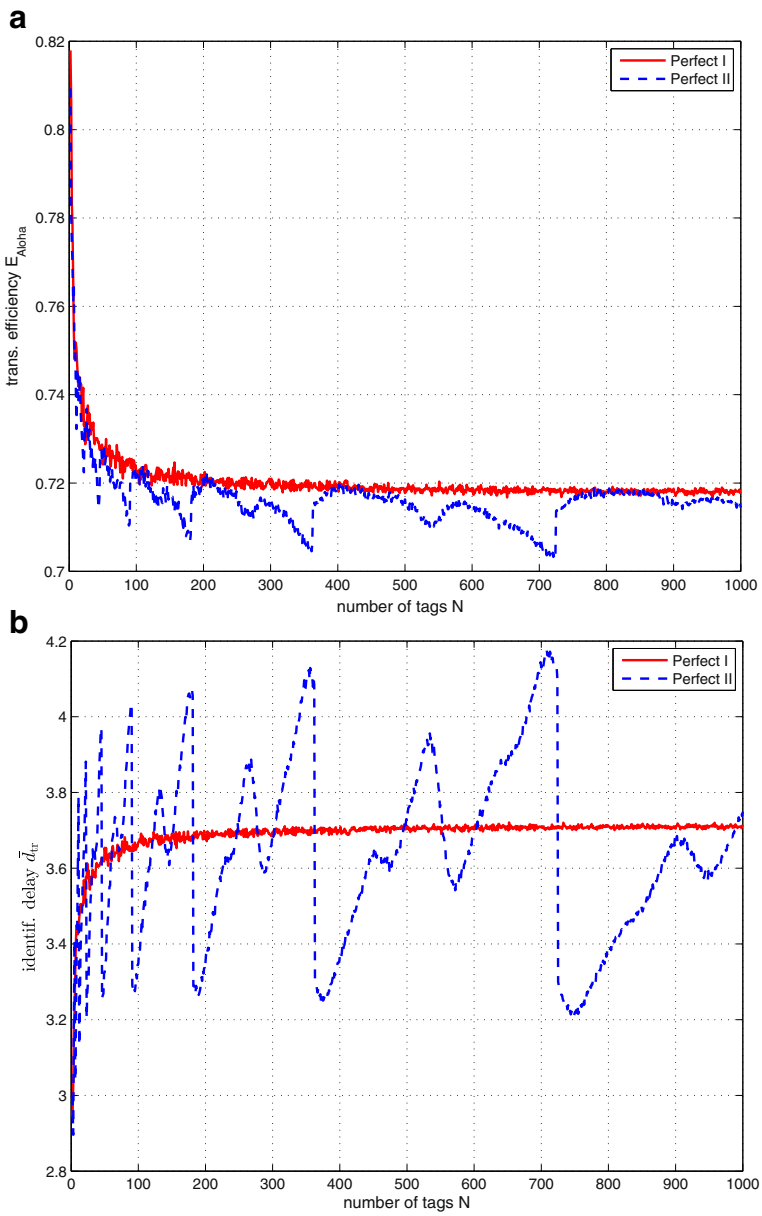


**Fig. 9** An illustration for the transmission efficiency and MID of Perfect I and Perfect II. **a** Transmission efficiency $E_{\text{Aloha}}$. **b** MID $\bar{d}_{\text{tr}}$

case of $x_{\text{EPC}} = 64$, the transmission efficiency of Perfect I decreases from 0.72 down to 0.718 when $N$ goes from 200 to 1000, while the transmission efficiency of Perfect II oscillates from 0.703 to 0.72 when $N \in [200, 1000]$. Correspondingly, the MID of Perfect I is below 3.72 TCs and the MID of Perfect II oscillates from 3.2 to 4.2 TCs.

Comparing Figs. 7, 8, and 9, we can see that the transmission efficiency of both AFSA2 and EPC Variant II is very close to that of Perfect I and Perfect II. For the case of $x_{\text{EPC}} = 64$, if we take 0.718 as the asymptotic value of the transmission efficiency of Perfect I and hence an upper bound for the transmission efficiency, then AFSA2 and EPC Variant II can achieve 92 % and 97 % of this upper bound, respectively.

Comparing Figs. 7 and 8, we can see that the setup of parameter $\Delta$ affects the performance of EPC Variant II marginally, but it affects the performance of EPC Variant I considerably. For example, when $N > 250$ and $x_{\text{EPC}} = 64$, the transmission efficiency of EPC Variant I is below 0.2 for the constant $\Delta$, which is even worse than that of the static framed Aloha with a proper frame size.

Figures 7 and 8 reveal that for the EPC Q-selection algorithm, the moment for the reader to adjust the frame size plays an important role. EPC Variant I and EPC Variant

II represent two extreme cases: EPC Variant II adjusts the frame size at the very beginning of a frame if idle or colliding transmissions, other than successful transmissions, happen at the first time slot, while EPC Variant I adjusts the frame size for the next layer when all the transmission contentions in the current layer are finished. It is conjectured that the transmission efficiency and MID can be well balanced if the moment for the reader to adjust the frame size is chosen in some time between the beginning and the end of a frame. We omit the further study in this direction since this is not the focus of this paper.

Since the computational burden of AFSA2 is negligible compared to that of AFSA1 (and even to the static Aloha), it is recommended to use AFSA2 with appropriate $R_{c0}$ (e.g., $R_{c0} = 0.5$ or 0.6) in RFID Aloha-based anti-collision design.

## 5 Conclusions

In this paper, we have discussed Aloha-based anti-collision algorithms for multi-tag RFID systems. It is shown that the static Aloha yields very poor performance in both transmission efficiency and MID. Therefore, we propose two adaptive frame size Aloha algorithms, namely AFSA1 and AFSA2, in which the frame size at the next layer is adaptively changed according to the real-time measured collision rate at current layer. Very light computational burden at the reader is needed: the reader needs only to measure the collision rate in the current layer and then to double or halve the frame size of the next layer accordingly. No additional computational burden is required at the tag side.

Simulation results show that AFSA1 and AFSA2 can significantly improve both transmission efficiency and MID of multi-tag RFID systems compared to those of the static Aloha; the transmission efficiency of AFSA2 is a little lower than but close to that of the EPC Variant II, and the MID of AFSA 2 is much smaller than that of EPC Variant II. It is worth noting that when the threshold of the collision rate is chosen to be 0.5 or 0.6, AFSA2 can maintain the transmission efficiency well above 0.65 for the case of a typical EPC code length of 96 bits and for the investigated range of tag population, i.e., from 2 to 1000, while keeping the MID below ten TCs. Besides, the transmission efficiency of AFSA2 is very close to the upper bound of the transmission efficiency obtained from the ideal system. Therefore, it is recommended to use AFSA2 in general.

In AFSA1 and AFSA2, the value of parameter $R_{c0}$ can be further optimized. Actually, two independent parameters can be freely chosen in this regard. The first one is the collision rate threshold $R_{c0}$ used to double the frame size, and another one is the collision rate threshold, say $R_{c1}$ instead of $\frac{1}{2}R_{c0}$, used to halve the frame size in AFSA2. An interesting research problem is to optimize these two independent parameters.

**Competing interests**
The authors declare that they have no competing interests.

**Acknowledgements**
The authors are grateful to the reviewers for their constructive comments, which helped improve the quality of this paper.

Received: 14 November 2015 Accepted: 8 March 2016
Published online: 12 April 2016

**References**

1. TF La Porta, G Maselli, C Petrioli, Anticollision protocols for single-reader RFID systems: temporal analysis and optimization. IEEE Trans. Mobile Comput. **10**(2), 267–79 (2011)
2. J Myung, W Lee, J Srivastava, TK Shih, Tag-splitting: adaptive collision arbitration protocols for RFID tag identification. IEEE Trans. Parallel Distrib. Syst. **18**, 763–75 (2007)
3. D-H Shih, P-L Sun, DC Yen, S-M Huang, Taxonomy and survey of RFID anti-collision protocols. Comput. Commun. **29**, 2150–66 (2006)
4. DK Klair, K-W Chin, R Raad, A survey and tutorial of RFID anti-collision protocols. IEEE Commun. Tutorials. **12**, 400–421 (2010)
5. EPCglobal, EPC radio-frequency identity protocols Generation-2 UHF RFID Specification for RFID air interface protocol for communications at 860 MHz–960 MHz, version 2.0.1 ratified (2015). Available http://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf
6. N Abramson, in *Proc. 1970 Fall Joint Computer Conf., AFIPS Press*. The ALOHA system—another alternative for computer communications, (New York, USA, 1970), pp. 281–5
7. L Liu, S Lai, in *Proc. Int. Conf. Wireless Communications, Networking and Mobile Computing*. ALOHA-based anti-collision algorithms used in RFID system, (Wuhan, China, 2006)
8. M-K Yeh, J-R Jiang, S-T Huang, Adaptive splitting and pre-signaling for RFID tag anti-collision. Comput. Commun. **32**, 1862–70 (2009)
9. F Schoute, Dynamic frame length ALOHA. IEEE Trans. Commun. **31**, 565–68 (1983)
10. JS Choi, H Lee, DW Engels, R Elmasri, in *2008 IEEE Int. Conf. on RFID*. Robust and dynamic bin slotted anti-collision algorithms in RFID system, (Las Vegas,USA, 2008), pp. 191–98
11. B Knerr, M Holzer, C Angerer, M Rupp, Slot-wise maximum likelihood estimation of the tag population size in FSA protocols. IEEE Trans. Commun. **58**, 578–85 (2010)
12. M Kodialam, T Nandagopal, in *Proc. 12th Annual Int. Conf. Mobile Computing and Networking*. Fast and reliable estimation schemes in RFID systems, (Los Angeles, CA, USA, 2006), pp. 322–33
13. S-R Lee, S-D Joo, C-W Lee, in *Proc. 2nd Annual Int. Conf. Mobile and Ubiquitous Systems Networking and Services*. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification, (San Diego, CA, USA, 2005), pp. 166–172
14. H Vogt, in *Proc. 1st Int. Conf. Pervasive Computing*. Efficient object identification with passive RFID tags, (Zurich, Switzerland, 2002), pp. 98–113
15. B Zhen, M Kobayashi, M Shimizu, Framed ALOHA for multiple RFID objects identification. IEICE Trans. Commun. **E88-B**, 991–99 (2005)
16. C Floerkemeier, in *2007 IEEE Int. Conf. on RFID*. Bayesian transmission strategy for framed ALOHA based RFID, protocols, (Grapevine, Texas, USA, 2007), pp. 228–235
17. NL Johnson, S Kotz, *Urn Models and Their Applications*, (Wiley, New York,1977)
18. F Zheng, T Kaiser, *Digital Signal Processing for RFID*, (Wiley, Chichester, 2016)
19. G Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun. **18**, 535–547 (2000)