Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Suffix-based double reader tag movement RFID anti-collision method

Haochen Lv[1,2,3], Xiaowu Li[1,2,3*], Lianyin Jia[1], Runxin Li[1], Jiaman Ding[1] and You Jinguo[1]

Abstract

At present, some existing tag anti-collision algorithms are used mostly in scenarios of fixed number of tags and they perform poorly in tag moving scenarios. Therefore, this paper proposes a double reader tag movement RFID anti-collision method based on suffix, and through the cloud platform based on cloud computing, the supply chain data sharing is realized. Tag moving scenarios usually use a single reader. The proposed employs double reader to expand the recognition range and also to prolong the recognition time, but there is the problem of repeated recognition. This article proposes a suffix-based bit-slot ALOHA (SBSA) algorithm, where this problem is solved and the time-slot is reduced. In view of the problem that a large number of tags entering into the recognition area of readers lead to the reduction of the recognition efficiency. This article proposes a suffix-based tag group bit-slot ALOHA (STGBSA) algorithm, tag grouping and tag estimation methods are proposed. Complexity analysis is deployed in this article, from the perspective of the tag loss ratio and identification efficiency, the performances of various anti-collision algorithms are evaluated and compared with those of the existing methods. The simulation results show that the method proposed in this paper is better than the existing methods.

Keywords: Radio frequency identification (RFID), Tag anti-collision, Tag movement, Double reader, Cloud computing

Introduction

In recent years, Internet of Things (IoT) has been widely used with the rapid development of a new generation of the information technology [1–4]. IoT is a kind of network, which uses information sensing equipment, such as radio frequency identification (RFID), infrared sensors, global positioning system, etc. It is used to connect objects to the internet for information exchange and communication according to the specified agreements, so as to achieve identification, positioning, tracking, monitoring, and management of items [5–7]. RFID is a very important part of the internet of things. The RFID system consists of a reader, a tag, a server, and a cloud platform. The reader uses radio waves to obtain information from the tags attached to items, which is transmitted to the server [8], the server uploads the tag information to the cloud platform. Getting the tag information is one of the key steps. However, when a common channel is used to communicate multiple tags with the reader at the same time, the problem of tag collision arises, which needs to be solved by an anti-collision algorithm.

At present, most of the anti-collision algorithms are employed in a static environment, where the tags are assumed to be still in the process of their identification. However, the identification of an express, such as logistics [9] and warehouse [10], is dynamic in real life, and studies related to it are few [11–13]. In this scenario, the reader recognizes tags that move along a fixed path, which is called the tag movement RFID system (TMRS).

The RFID tag anti-collision methods include mainly tree-based deterministic algorithms [14, 15] and ALOHA-based random algorithms [16–19]. In the tree-based deterministic algorithms, the recognition rate can

* Correspondence: lxwlxw66@kmust.edu.cn; weiyouhuangpy@163.com
[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China
[2] Key Lab of Computer Technologies Application of Yunnan Province, Kunming 650504, China
Full list of author information is available at the end of the article

reach 100%. However, this type of protocol has a large delay and it is not suitable for real-time recognition in a dense tag environment. In situations where the tags move quickly in or out of the recognition range of the reader, the ALOHA-based random algorithms are not suitable to be considered as the tree-based deterministic algorithms.

Some scholars proposed ALOHA-based algorithms. Wong [20] proposed a bit slot ALOHA algorithm based on bit sequences. In DSFA algorithms, Chen [21] proposed a method by the Maximum a Posteriori (MAP) to estimate the tag number, HajMirzaei [22] proposed an efficient method to estimate the lower bound of collided tags by Manchester coding, Wang [23] proposed the secant-iteration-based adaptive and dynamic frame slotted Aloha method (SIADA), Yang [24] proposed a low complexity dynamic frame slotted Aloha (LC-DFSA) anti-collision algorithm. The algorithm proposed in this paper will be compared with the algorithms proposed in recent years, reflecting the good performance of the algorithm in TMRS.

The terms used in this article are defined below:

Tag flow-A group of tags are scattered on a conveyor belt and moved into or out of the signal area of the reader through the conveyor belt.

Tag moving density (TMD)-The TMD unit represents the number of tags per slot (tag/slot). For example, TMD = 0.8 tag/slot means that there are 0.8 tags in a slot entering the signal area. Defining the unit of TMD is conducive in analyzing the performance of the dynamic RFID system. Since the reader takes a slot to successfully read the complete information of a tag, TMD is greater than 1 tag/slot, which means a high density.

We analyze the tag movement RFID system based on passive tags. It obtains information about goods by making readers (usually a single reader is used) to identify the passive RFID tags on the conveyor belt. However, the recognition range for a single reader is limited. And the conveyor belt needs to maintain a certain speed determined by the power of the used motor and the governor. The torque is not large enough at a low speed. The conveyor belt becomes unable to move whenever a weight is added to it. Further, the waveform of the output voltage is not good if the speed of the governor is low, which is also harmful to a motor running at a low speed for a long time [25, 26]. This shortens the recognition time, leading to missing of tags. The design of double reader model is proposed, which expands the recognition range and prolongs the recognition time. But the interference signal area between the readers becomes much larger than the recognition area. Also, there is a problem with a hidden terminal if it is within the signal interference range of a reader. When the recognition areas of the readers are close to each other, interference between them occurs, whereas they need to work separately. In [12], a reader can determine whether the tag is unread by the previous reader. But, if the tag is a passive one and there is any intermediate power cut, the reader will be unable to determine whether the tag is recognized or not. It will identify the already recognized tags again, and thus will waste a lot of slots on repeated identification.

There is a problem of repeated recognition between readers. This paper proposes a suffix-based bit-slot ALOHA (SBSA) algorithm. In this method, this problem can be solved. Hence, the wastage of the slot cost on repeated identification is saved, and also the slot cost of collided slots is also reduced.

When the tag movement density is high, a large number of tags enter into the signal area of the reader at the same time. This results in increasing tag collisions and makes it difficult to read tags successfully. In order to solve this problem, a suffix-based tag group bit-slot ALOHA (STGBSA) algorithm is proposed in this paper, which controls the number of tags to be read at a time, and thus reduces the likelihood of collisions.

The rest of this article is arranged as follows. In the "Double reader model and cloud platform architecture" section, the double reader model and cloud platform architecture are introduced. In the "Algorithm design" section, the correlation algorithm is introduced. In the "Simulation results and analysis" section, the simulation results and analysis are summarized. Finally, the paper is summarized In the "Conclusions" section.

Double reader model and cloud platform architecture

In actual applications, such as the scanning of shipments in an express warehouse, the shipments at the beginning are scattered on a conveyor belt. The UHF reader [27] is selected as it has a very long working distance (its working frequency is 862 (902) ~ 928 MHz, and it fully supports electronic tags that comply with the EPC CLASS1 G2 and ISO18000-6B standards [28]. Its working distance is usually greater than 1 m, typically 4 ~ 6 m. Two readers are built and fixed on the upper end of the conveyor belt. In view of the working frequency and signal interference, the readers read information sequentially. When the movement of tags starts, reader 1 runs. There are infrared sensors between the recognition areas of the two readers. When the induction line of the infrared sensor senses the arrival of a tag, the second reader starts and the first reader closes. The flowchart is shown in Fig. 1.

The definition of a double reader tag movement RFID system is that it uses two readers to sequentially identify

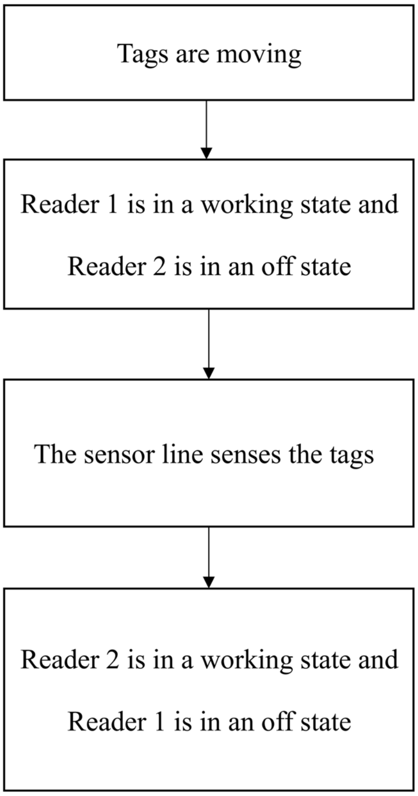

Tags are moving
Reader 1 is in a working state and Reader 2 is in an off state
The sensor line senses the tags
Reader 2 is in a working state and Reader 1 is in an off state

Fig. 1 The flowchart of double reader

a batch of moving tags in a tag stream environment. This kind of RFID system is called a double reader tag movement RFID system (DRTMRS). Accordingly, this paper builds a DRTMRS model as shown in Fig. 2.

The steps of building the model are as follows: Firstly, the length and width of the tag stream are set to be Lx and Ly, respectively, and N new tags are randomly distributed among tag stream. Secondly, the signal radius of reader 1 is set to be $R1_{radius}$ and that of reader 2 is to be $R2_{radius}$. There are induction lines between the recognition areas of the two readers. Then, the tag speed is set to be V_{step}. Two readers send the acquired label EPC information to the local server.

The local server uploads the tag EPC information to the corresponding regional cloud platform through EPC information service (EPCIS) to complete EPC-related data storage. Cloud platforms in each region form a unified cloud platform to realize data sharing among supply chains. The cloud platform is built according to the idea of cloud computing [29, 30], and the demand for cloud resources can be adjusted in real time according to the information flow of the supply chain. The cloud platform architecture is shown in Fig. 3. This article is mainly about the double reader method for preventing tag collision in the cloud platform, and it talks only a little about the cloud platform.

Algorithm design

BSA algorithm

The bit-slot ALOHA algorithm [20] is implemented based on a bit sequence. The sequence consists of L bit slots. A bit slot is randomly selected for a tag, which is set to be 1 and all other bit slots are set to be 0. The reader sends an inquiry command to the tag, which randomly generates a bit sequence and sends it to the reader. The reader detects the collided bits in all the bit sequences it receives. If a bit collided, it indicates that the bit slot is selected by at least one tag as shown in Fig. 4. The reader records the collided bits, and the EPC code of the tag corresponding to the collided bits is then asked in turn. If only one tag responds, then the tag is read successfully, otherwise, its recognition is failed.

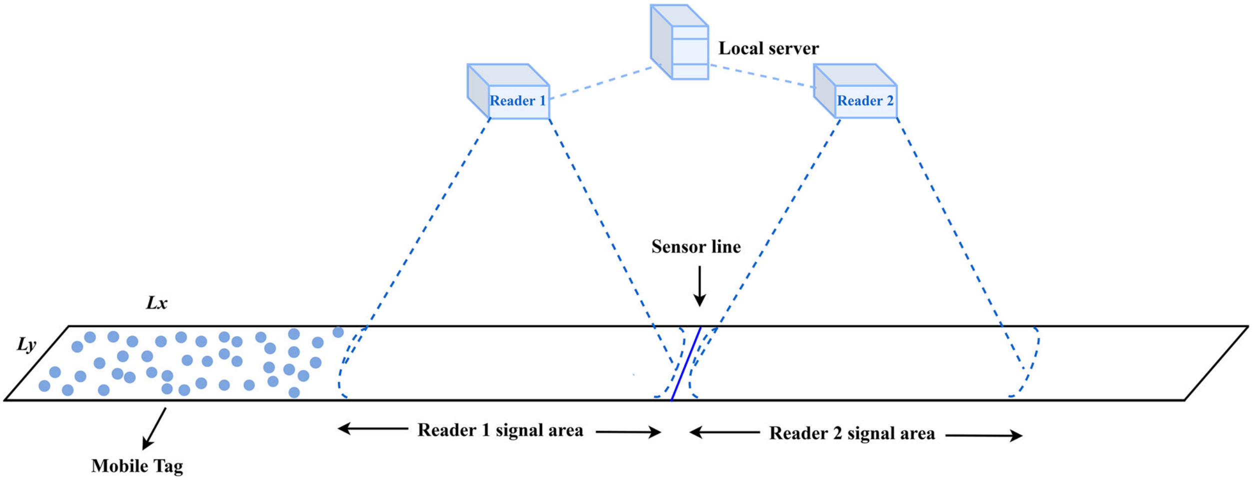


Fig. 2 Double reader model

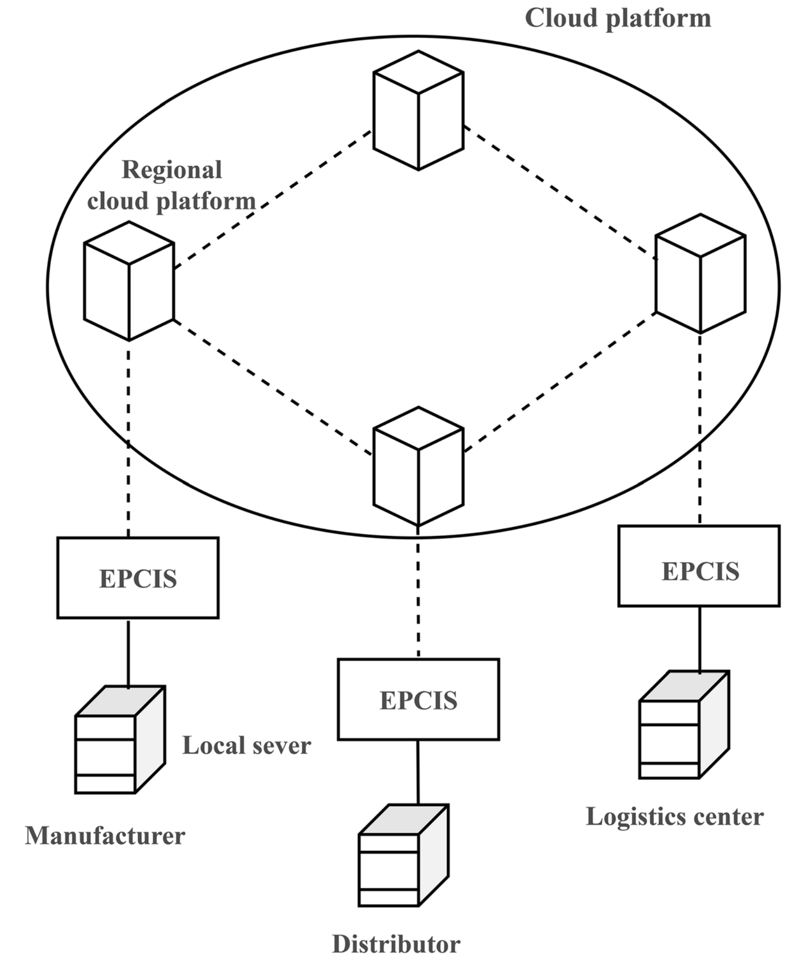


Fig. 3 Cloud platform architecture

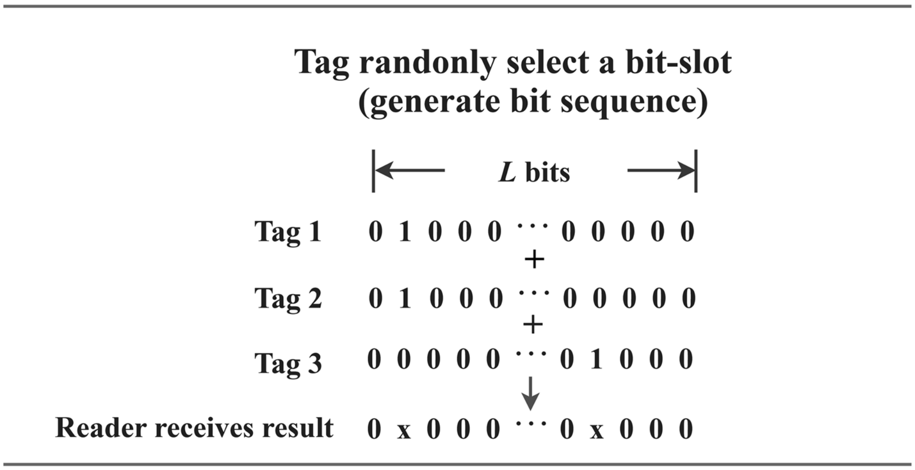

Tag randonly select a bit-slot
(generate bit sequence)
L bits
Tag 1 0 1 0 0 0 ··· 0 0 0 0 0
+
Tag 2 0 1 0 0 0 ··· 0 0 0 0 0
+
Tag 3 0 0 0 0 0 ··· 0 1 0 0 0
Reader receives result 0 x 0 0 0 ··· 0 x 0 0 0

Fig. 4 Generation of bit sequences

In the bit-slot ALOHA algorithm, the bit sequence includes L bit slots. Assume that the number of tags in the signal area of the reader is N. Then, the probability of a tag being selected for a certain bit slot is $p = 1/L$.

Thus, the probability of the slot selected only for one tag (success slot) is:

$$P_{succeed} = C_N^1 p^1 (1-p)^{N-1} \tag{1}$$

The expected value of the total number of slots (success slots) selected only for one tag is:

$$L_{succeed} = L * P_{succeed} \tag{2}$$

The probability of the slot selected for more than one tag is:

$$P_{collide} = 1 - C_N^0 p^0 (1-p)^N \tag{3}$$

The expected number of collided bit slots is:

$$L_{collide} = L * P_{collide} \tag{4}$$

Set an inquiry of the reader and the corresponding response of the tag as an inquiry cycle (it costs one slot). The tag responds to the bit slot and fully recognizes it as a recognition cycle (it costs one slot). If the reader detects that $L_{collide}$ bit slots collided, then $L_{collide}$ recognition cycles are needed to read the EPC code of each tag. The efficiency of identification is defined as the average number of tags identified in each slot. Then, the identification efficiency β can be expressed as:

$$\beta = \frac{L_{succeed}}{L_{collide} + L_{succeed} + 1} \tag{5}$$

As only the collided bit slots are read and there is no free slot in the algorithm, it is more efficient compared to the frame slotted ALOHA algorithm.

Introduction of suffix

The EPC coding can realize binary 96-byte information storage as shown in Fig. 5. Each tag has a unique code. When processing a tag, only a part of the EPC code (serial number, 36 bytes), namely the suffix, is processed as the entire EPC code is very long. Reading all the information about a tag requires a slot, and the cost of read the suffix is about 37.5% (36/96) of reading all the information.

As only a part of the EPC code is selected, the suffix of the tag may be repeated. When the number of tags to be identified is N and the length of the suffix of a tag is L_{suffix}, the non-repetition probability of the suffix is:

$$P_{non-repetitive} = \frac{2^{L_{suffix}}!}{\left(2^{L_{suffix}} - N\right)!} * \frac{1}{\left(2^{L_{suffix}}\right)^N} \tag{6}$$

Therefore, the repetition probability of the suffix is:

$$P_{repetitive} = 1 - P_{non-repetitive} \tag{7}$$

When the length of the suffix remains unchanged and the number of tags to be identified changes, the repetition probability of the suffix is as follows:

When $L_{suffix} = 36$ and $N = 1000$, then $P_{repetitive} = 7.268 * 10^{-6}$;

When $L_{suffix} = 36$ and $N = 5000$, then $P_{repetitive} = 1.818 * 10^{-4}$;

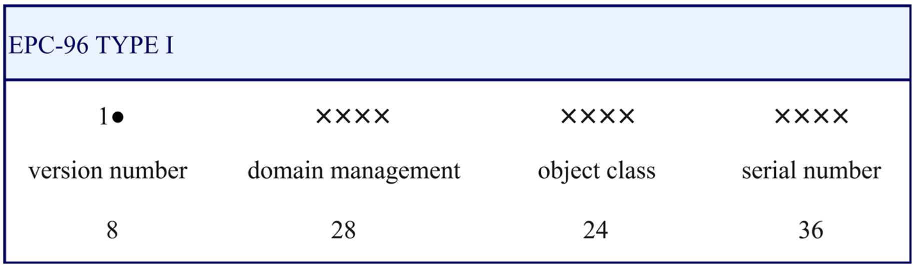
When $L_{suffix} = 36$ and $N = 10000$, then $P_{repetitive} = 7.272 * 10^{-4}$.

Although the repetition probability of the suffix is very small for the length of the suffix is 36 bytes, it can be seen that the repetition may still take place irrespective of the number of tags. In this situation, the suffix reprocessing (SR) becomes necessary.

For using the suffix, two suffix tables are required, namely the global suffix table (GST) and the bit suffix table (BST). GST is used to store the suffix of all the recognized tags, while BST is used to record the suffix of the tags received by the reader and appeared in GST during a single reading. The hash table data structure and hash lookup store and lookup suffix respectively.

Construction and application of the SBSA algorithm

The combination of the suffix and the BSA algorithm is called the suffix-based bit slot ALOHA (SBSA).



EPC-96 TYPE I

1●	××××	××××	××××
version number	domain management	object class	serial number
8	28	24	36

Fig. 5 EPC coding

As shown in Algorithm. When the reader sends an inquiry command to the tag in the signal area, the tag randomly generates a bit sequence, and then sends it to the reader. The reader records all the collided bits B and their total number $L_{collide}$, and then inquires the tag for its suffix in turn. The tag sends the suffix and its length L_{suffix} to the reader at the beginning of its bit slot, and there are two types of responses:

1) Multiple tags response: multiple tags send suffix at the same time, collision of suffix will occur consuming slots of the suffix length. Identification tag failure, these tags will participate in the next round of identification cycle.
2) Single tags response: the suffix of the individual response compares the suffix with that of the identified tag (stored in the global suffix table):
   2.1)If the suffix appears in the GST, we call the corresponding tag as a similar identification tag. For this type of tag, the reader stores the suffix in the BST. The remaining portion of the EPC code of a similar identification tag is sent to the reader in a certain proportion during the suffix reprocessing. The reason is that the suffix of a similar identification tag may come from an identified or an unidentified tag. If the suffix is derived from an identification tag, it means that the signal area of the reader covers the tag again, otherwise it means that there is at least one tag with the same suffix. The suffix in the BST is processed later.
   2.2)If the suffix does not appear in the GST (it means that the tag has not yet been recognized), the tag immediately sends the remaining portion of its EPC code to the reader and puts the suffix in the GST.

The suffix reprocessing (a certain proportion of the suffix should be extracted from the BST for further identification) after all the collided bits slots are over. This ratio is called the suffix reprocessing proportion (SRP). The purpose is not to miss unidentified tags having the same suffix with identified tags. Summary of the SBSA algorithm is as follows: The suffix is used to determine whether the tag is identified and also to reduce the slot. The slot spent by the reader to read the suffix is about 37.5% (36/96) of that spent to read all the information. The identification efficiency β_{suffix} is expressed as:

$$\beta_{suffix} = \frac{L_{succeed}}{(L_{collide} - L_{succeed})*0.375 + L_{succeed} + 1} \tag{8}$$

Algorithm Operation in SBSA

```
Set $L_{suffix} = 36, i = j = k = 1, B, L_{collide}$;
Send Query command;
Receive the bit sequence of the tag;
Record all the collided bits $B(1, L_{collide})$;
while $i < L_{collide}$ do
  Query $B(1,i)$ to tag;
  if only one tag responds then
     receive $L_{suffix}$ suffix;
    if the suffix exists in GST then
       the suffix is put in BST;
    else
       the suffix is put in GST;
       send read remaining EPC command to the reader;
       receive the remaining EPC code of the tag;
    end
  else if tags collide then
     spend $L_{suffix}$ slot;
  end
  $i++$;
end
Execute Suffix Reprocessing
if there exists at least one suffix in BST then
  while $k = 1$ do
    take a suffix from BST;
    send the remaining EPC command to the reader;
    receive the remaining EPC code of the tag;
      if $j <$ the number of suffix in BST * SRP then
       $k = 1$;
      else
       $k = 0$;
      end
     $j++$;
    end
end
```

Table 1 The theoretical relationship between collision and tag number

N_{theory}	1	10	50	100	200
$LT_{collide}$	1	9.66	41.52	69.58	101.33

STGBSA algorithm

In view of the problem that the recognition efficiency will be greatly reduced when a large number of tags will enter into the recognition area of the reader at the same time, the Suffix based Tag Group Bit-slot ALOHA (STGBSA) algorithm is proposed.

Estimated number of tags

The reader sends n query commands to the tag, and then the tag randomly generates a bit sequence consisting of L bit slots and sends it to the reader. The average of the numbers of collision bits obtained separately against n query commands is taken as the actual number of collision bits, and it is denoted by $LP_{collide}$. Theoretically, the relationship between the numbers of collision bits $LT_{collide}$ and the number of tags N is determined by Eq. (4), when $L = 128$, then as shown in Table 1. By comparing the actual number of collision bits with the theoretically expected value, the estimated number of tags $N_{estimate}$ is estimated as the theoretical number of tags N_{theory} when the error is minimum. The specific formula is given below:

$$\varepsilon = \min_{N} |(LT_{collide}) - (LP_{collide})| \tag{9}$$

When the error is minimum, N_{theory} becomes $N_{estimate}$.

Figures 6 and 7 show, respectively, the mean error and the maximum error between the estimated number of tags and the actual number of tags obtained through 100,000 Monte Carlo simulations. It can be seen that the larger the value of n, the smaller the mean error, maximum error and variation. Considering the time cost, $n = 5$ is taken as the number of times the query command is sent. This method can effectively estimate the number of tags with a fewer slots.

Tag grouping method

In the tag movement RFID system, tags keep entering into the signal area of the reader. Li [31] showed that tags can be grouped according to their entry time into the signal area. The procedure is to assign the same sequence number (SN) to all the tags entering into the signal area during the tag identification process. In this way, the tags, entering into the signal area at the same time period, get the same SN and belong to the same tag group. The sequence numbers of the tags, entering into the signal area at different time periods, are different. The SN is set to be 1, 2, 3, ..., etc. This is called the tag grouping method, which implements the tag group sequencing (TGS) as shown in Fig. 8. We can see that tags for a period of time are grouped into the same tag group. The reader sends a command with a SN, and the tag having that SN can respond to the reader. The time interval of the grouping operation is called the time interval of sequencing (TIS). It can be set as needed. Since the grouping also costs one slot, the identification efficiency $\beta_{grouping}$ can be calculated as follows:

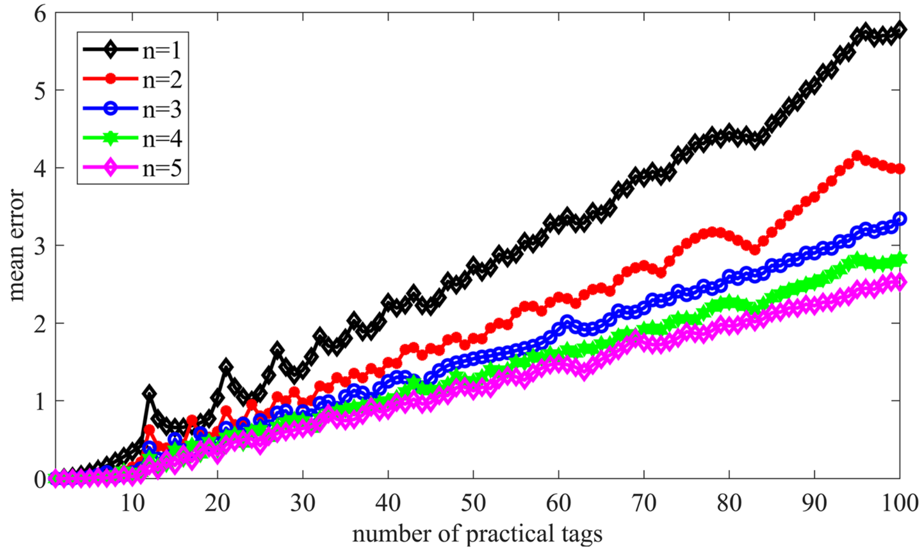


Fig. 6 Mean errors for different n

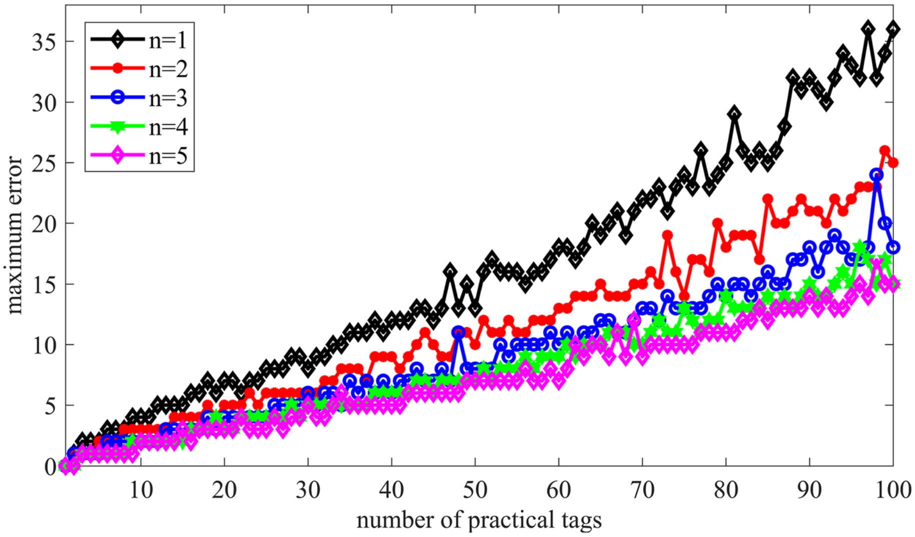


Fig. 7 Maximum errors for different n

$$\beta_{grouping} = \frac{L_{succeed}}{(L_{collide} - L_{succeed})*0.375 + L_{succeed} + 2} \tag{10}$$

Algorithm construction and application

The flow of the STGBSA algorithm flow as follows: The tags begin to enter the signal area of the reader. Firstly, the initial time interval is set to $T_{initial}$. The tags are assigned to group G_1 after grouping. The estimated tag method is used for the tag group G_1 to determine the estimated tag number $N_{estimate}$. The sequential time interval of the tag group T is determined by the initial time interval $T_{initial}$, the estimated number of tags $N_{estimate}$ and the number of suitable tags found in the tag group $N_{suitable}$ through the recognition efficiency. The calculation formula is given by Eq. (11). In “Suitable tags for readers” section, the numbers of suitable tags for reader 1 and 2, and the identification using the SBSA algorithm will be discussed. The tags entering into the signal area are grouped according to the sequential time interval of the tag group T, and assigned to groups $G_i (i = 2, 3, 4, etc.)$. They are also identified by using the SBSA algorithm. Figure 9 provides the flow chart of the STGBSA algorithm.

$$T = T_{initial} * \left(1 + \frac{(N_{suitable} - N_{estimate})}{N_{suitable}}\right) \tag{11}$$

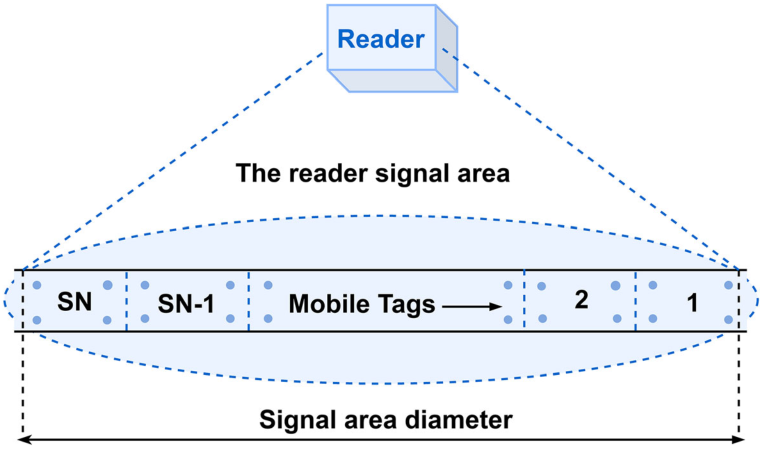


Fig. 8 Tag grouping mode

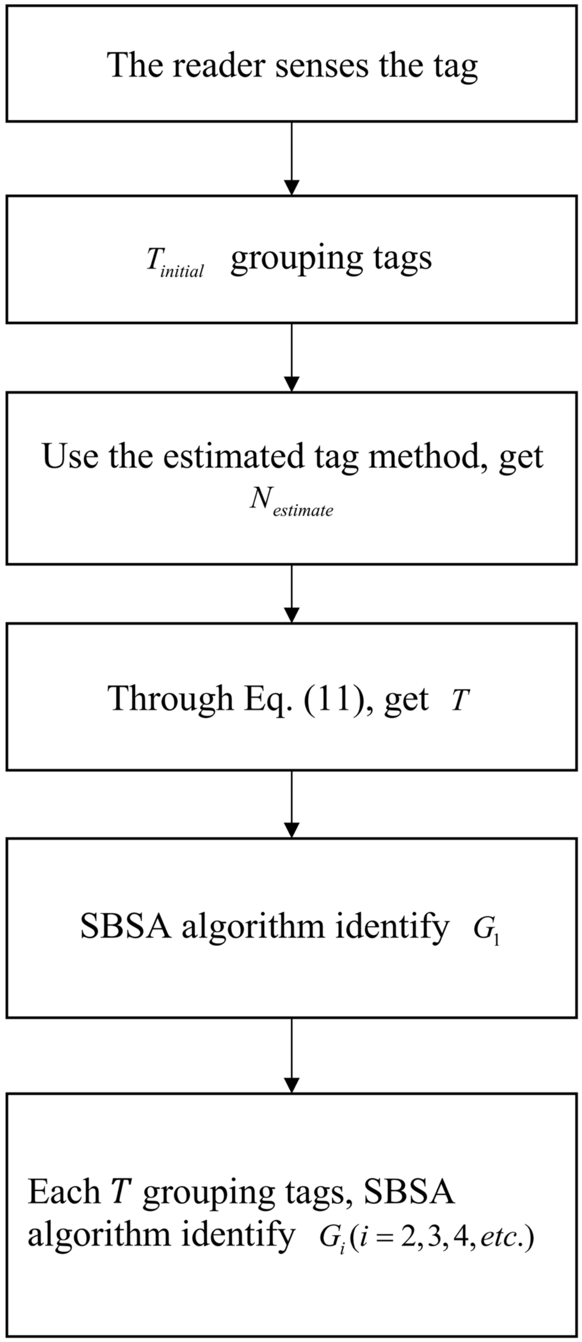

The reader senses the tag
$T_{initial}$ grouping tags
Use the estimated tag method, get $N_{estimate}$
Through Eq. (11), get T
SBSA algorithm identify G_1
Each T grouping tags, SBSA algorithm identify $G_i(i=2,3,4,etc.)$

Fig. 9 The flowchart of the STGBSA algorithm

Suitable tags for readers

Number of suitable tags for reader 1: In the STGBSA algorithm, reader 1 recognizes a group of tags only once even if it is not fully recognized, which is effective for large TMD. It also uniformly distributes the remaining new tags recognized by reader 2. The single recognition efficiency is considered here. The relationship between the recognition efficiency of reader 1 and the number of tags can be obtained by using Eq. (10). The simulation of the single recognition of the tag group is shown in Fig. 10. When $L = 128$ and $N = 34$, the recognition efficiency of the system reaches its maximum value, which is about 88.54%. The best performance is obtained at around $N = 34$. A better performance can be obtained if reader 1 can always be made to work in this interval.

Number of suitable tags for reader 2: In the STGBSA algorithm, reader 2 may need to recognize the same group of tags multiple times until it is fully recognized and then to proceed to recognize the next group of tags, so as to ensure that no tag is missed. Therefore, the recognition efficiency should consider multiple recognitions. The relationship between the recognition efficiency of reader 2 and the number of tags can be obtained by using Eq. (10). The simulation of the full recognition of a tag group is shown in Fig. 11. When $L = 128$ and $N = 42$, the recognition efficiency of the system reaches its maximum value, which is about 88.17%. The best performance is obtained at around $N = 42$. A better performance can be obtained if reader 2 can always be made to work in this interval.

It should be noted that reader 1 recognizes all new tags, and hence no suffix judgment is required here. The suffix and the remaining EPC codes are directly read by the successful slot. Reader 2 also needs to do suffix judgment.

Complexity analysis

Computational complexity is shown in Table 2. Wang estimate the number of initial tags by secant iteration, and used two probing frames to eliminate the Pseudo-solution, which has certain computational complexity. HajMirzaei proposed an efficient method that estimates the lower bound of collided tags (ET) by Manchester coding, whose value closest to the number of unidentified tags are determined according to the difference between the optimal and calculated numbers of collided slots in each cycle. The formula used in the method is not complicated. Chen used the polynomial distribution of repeated independent tests to simulate the exact probability of empty slots, success slots, and collided slots, whose formulas are computationally very intensive. Yang estimated the range of tag numbers according to the last frame size, the number of successful slots and the ratio of idle slots. Then the optimal frame size can be calculated. All the demarcation points and border values of LC-DFSA can be fairly simple. The SBSA and STGBSA algorithms used the hash table data structure and hash lookup store and lookup suffix, respectively, for which only a few computing resources are needed. But the STGBSA algorithm also involves the counting of the number of collided bits, estimating of the number of tags, and grouping of tags. The low-complexity

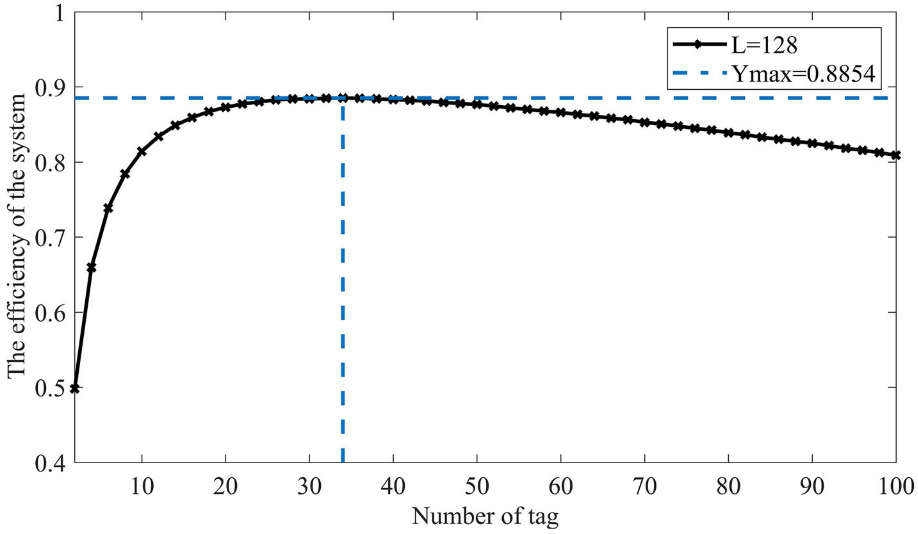


Fig. 10 Single identification of system efficiency versus the number of tags

HajMirzaei, Yang and SBSA were taken as examples for simulation, when $L = 128$, frame length interval $L_{\mathrm{int}erval}$ (2–512), as shown in Fig. 12. It can be seen that the algorithm complexity of SBSA algorithm is not the lowest in static recognition, and some time are spent in generating bit sequences. The Manchester coding collision of HajMirzaei algorithm increases the time slot overhead, and Yang algorithm has the lowest algorithm complexity due to its simple calculation formula.

Simulation results and analysis

In order to verify the applicability and effectiveness of the STGBSA algorithm proposed in this paper. HajMirzaei, Yang algorithms as well as the SBSA and STGBSA algorithms are applied to the double reader tag movement RFID system. The tag loss ratio (TLR) and the identification efficiency (IE) are used as the performance indicators to evaluate the performances of the considered anti-collision algorithms. The TLR refers to the ratio of the number of tags lost to the total number of tags, and the IE refers to the ratio of the number of identification tags to the total slot cost (the tag flow enters and leaves the double reader area). Finally, the influence of the ratio of suffix reprocessing is discussed. MATLAB is used in the simulation experiment. In the simulation, the tags are considered to be uniformly distributed. The detailed information of the parameters is given in Table 3.

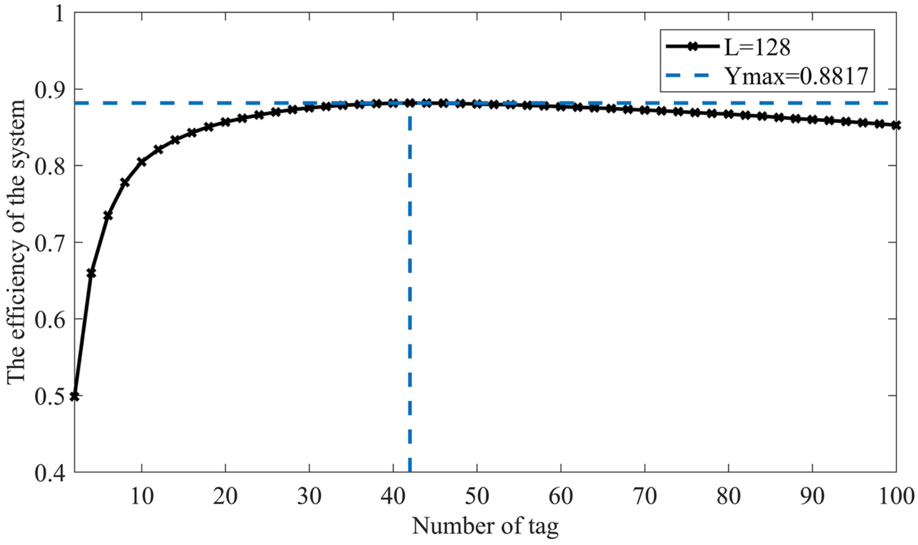


Fig. 11 Multiple identifications of system efficiency versus the number of tags

Table 2 Complexity comparison

Algorithms	Computational complexity
Wang [23]	Normal
HajMirzaei [22]	Low
Chen [21]	High
Yang [24]	Low
SBSA	Low
STGBSA	Normal

Analysis and comparison of simulation results of TLR and IE

Figures 13 and 14 show that the number of tags increases gradually. The number of tags changes from 1000 to 3000 (that is, the tag moving density TMD is less than 1 tag/slot). In HajMirzaei algorithm, the tag loss ratio increases sharply and the growth rate of the identification efficiency slows down when the number of tags is greater than 1700. When the number of tags reaches 2200, the identification efficiency reaches its peak value of 0.2105 and decreases thereafter. Since estimate the number of tags is more accurate, the performance of Yang algorithm is better than that of HajMirzaei algorithm. When the number of tags is greater than 1800, the tag loss ratio also increases sharply, and the growth rate of the recognition efficiency becomes slow. When the number of tags reaches 2300, the identification efficiency reaches its peak value of 0.221 and decreases thereafter. When the number of tags is 3000, the tag loss ratios of HajMirzaei, Yang, the SBSA and STGBSA algorithms are 0.4587, 0.4317, 0 and 0, respectively, and their recognition efficiencies are 17.51%, 19.04%, 33.33% and 33.33%, respectively. Both HajMirzaei and Yang lose tags and miss reading tags under a low density, IE reaches its own peak. The SBSA and STGBSA algorithms have no tag loss, and the growth rates of their recognition efficiencies remain unchanged, which reflect the efficiencies of the algorithms. The proposed algorithms are highly applicable to the tag movement RFID system.

Simulation results and analysis of TLR and IE under high density

Entering of a large number of tags into the signal area of the reader at the same time results in collision of increased number of tags, and the tag moving density (TMD) greater than 1 tag/slot, which is a high density. HajMirzaei has the highest TLR and the lowest IE under a low density, we need not to talk about it under high density. Figures 15 and 16 show that in the case of a high density, the BSA algorithm always has tag loss, the TLR keeps increasing, and the IE keeps decreasing. The SBSA algorithm starts to lose tags when the TMD is 1.12 tag/slot and the TLR is 0.2%. After that, the TLR gradually increases, making it difficult to handle the case under a higher density. However, the SGSBA algorithm solves the problem that the recognition of a large number of tags at the same time leads to increased tag collisions in the grouping method. The tag loss occurs only when the TMD is 1.30 tag/slot and the TLR is 0.12%. At this point of time, the TLR of the SBSA algorithm

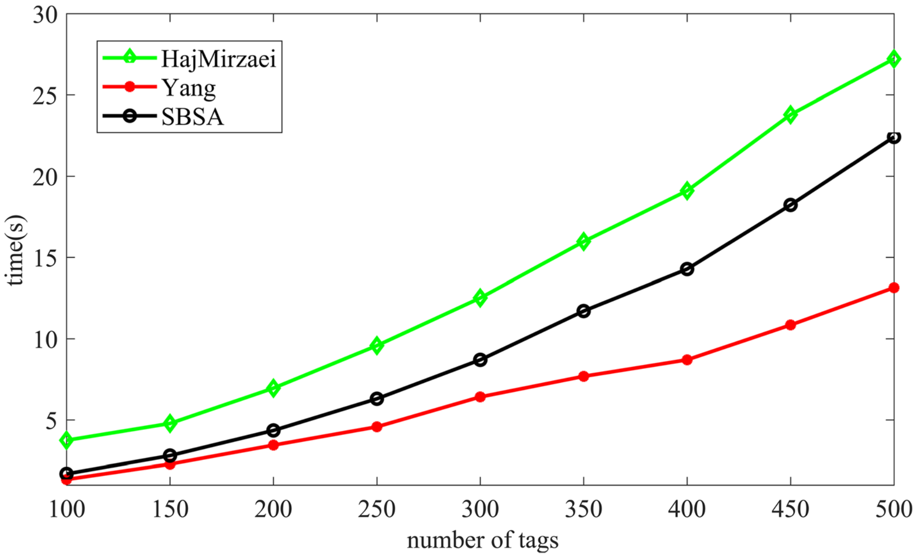


Fig. 12 time of different algorithms

Table 3 Default attribute values used in the simulation

Attribute	Value/Setting	Meaning and explanation
Lx	6000 mm	The length of the tag flow
Ly	600 mm	The width of the tag flow
$R1_{radius}$	3000 mm	The signed radius of the reader 1
$R2_{radius}$	3000 mm	The signed radius of the reader 2
V_{step}	2 mm/slot	The moving velocity of the tag
L	128	Bit sequence length
L_{suffix}	36	Suffix length
SRP	10%	Suffix reprocessing proportion
L_{interval}	2–512	The frame length interval of the Dynamic Frame Slotted ALOHA-based algorithms

is 19.25%. When the TMD is 1.5 tag/slot, the TLR values under the BSA, SBSA and STGBSA algorithms are 0.7532, 0.3709 and 0.12, respectively, and their IE values are 12.34%, 31.46% and 44.00%, respectively. It can be seen that when the tag movement density is high and the number of tags is large, the SGSBA algorithm is able to reduce the possibility of the missing of reading tags and to significantly improve the performance of the system.

Simulation results and analysis of the influence of SRP

When the suffix length L_{suffix} is 36 and the number of tags N is 4000, the suffix repetition probability $P_{repetitive} = 1.638 * 10^{-4}$. Figure 17 shows that the number of losing tags reduces when the SRP increases. This is because, the larger the SRP, the more suffix needs to be processed by consuming more slots. However, if the length of the suffix is large enough, the probability of repeating suffix is inherently low. However, the increased proportion does not have much effect on it. Note that it is useful to increase the proportion if the suffix repeating probability is relatively high.

Conclusions

The tag movement RFID system usually uses a single reader. However, the recognition range of a single reader is limited, which results in a short recognition time and missing of reading tags. In view of this problem, the design of double reader is proposed in this paper. An SBSA algorithm is provided, which uses the suffix to reduce the collision of slots and the slot cost of repeated identification between the readers. The STGBSA algorithm is

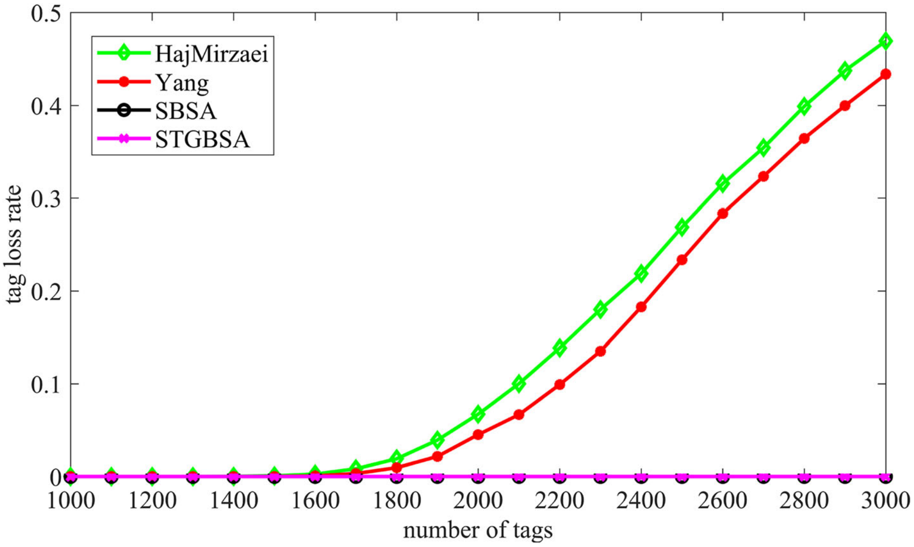


Fig. 13 TLR of different algorithms

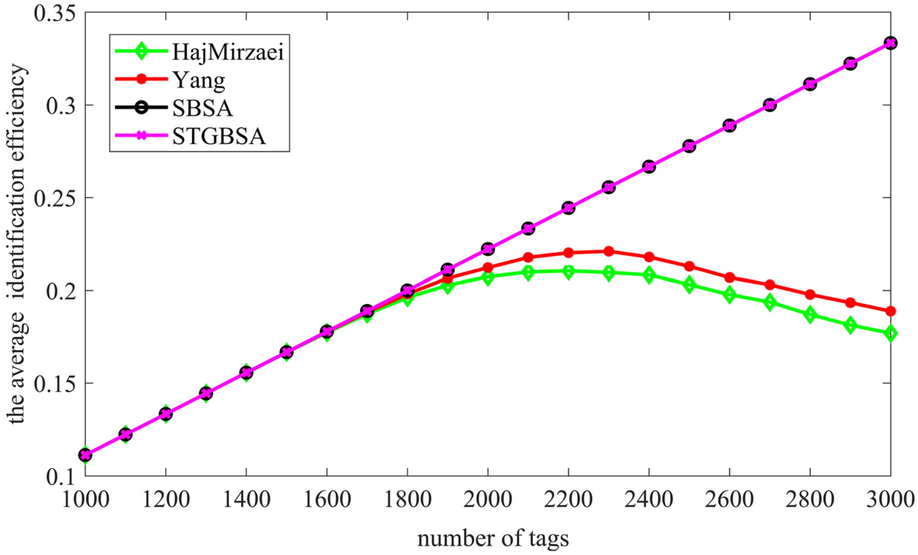


Fig. 14 IE of different algorithms

also proposed, which groups the estimated number of tags to solve the problem that a large number of tags entering into the signal area of readers at the same time lead to increasing tag collision and difficulty in successfully reading tags. Simulations show that as the number of tags increases, the performances of the proposed SBSA and STGBSA algorithms are greatly improved compared to other algorithms. Especially when the TMD is a high density, the STGBSA algorithm can ensure that no tag is missed in the tag grouping method (which the tags of each tag group successively). It solves the problem, which is left out in the ALOHA algorithm, that the tags are read randomly, resulting in missing of reading tags. Therefore, the design and the algorithms proposed in this paper have certain application values in the movement RFID systems. In the future, the algorithm will be optimized to consider the impact of change the suffix length and bit sequence length.

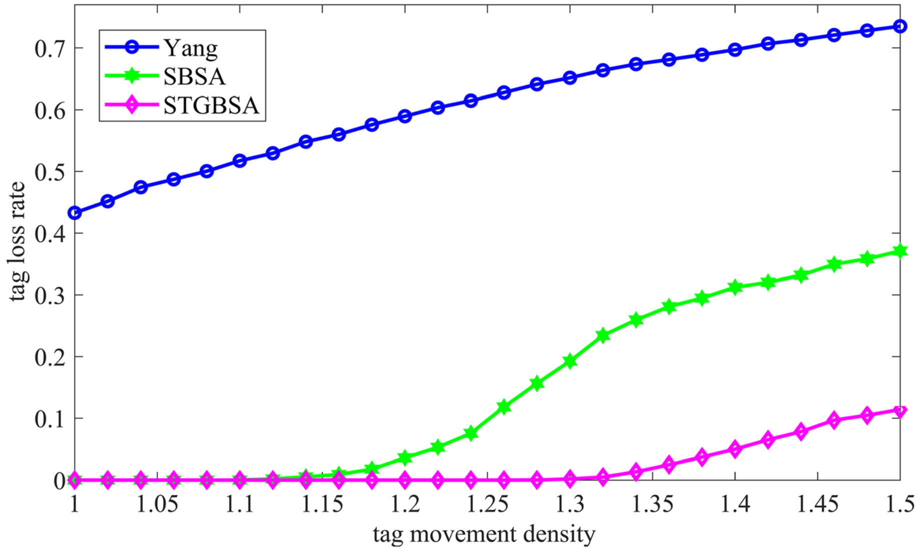


Fig. 15 Comparison of TLR at high density

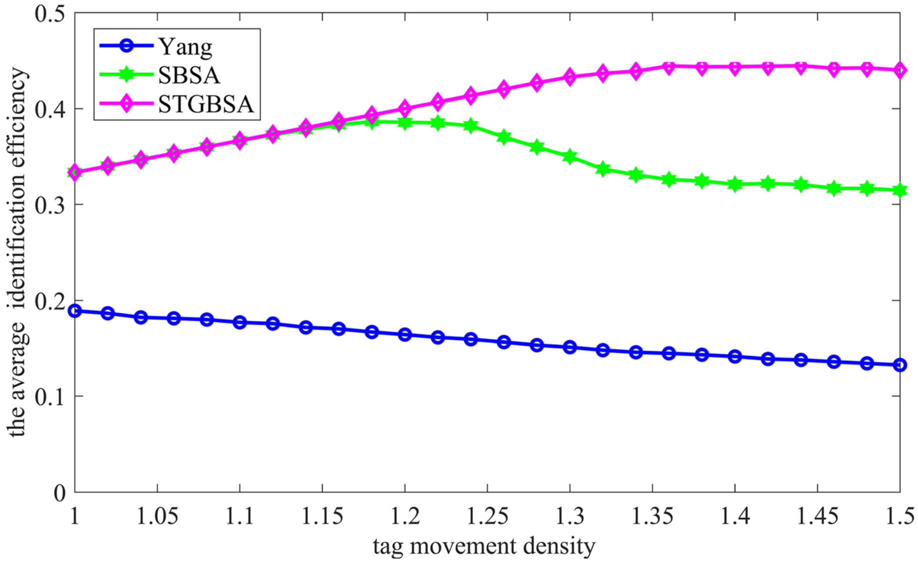


Fig. 16 Comparison of IE at high density

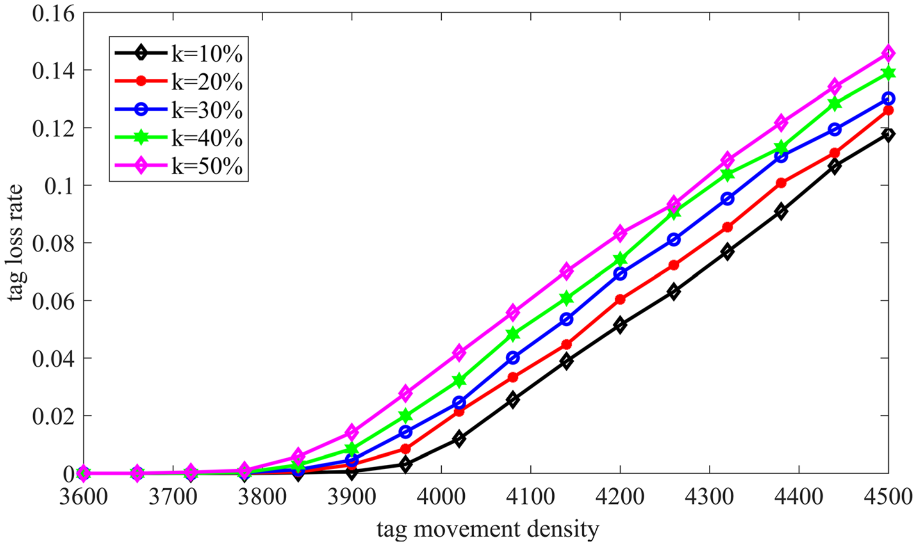


Fig. 17 Comparison of TLR between different SRP

Acknowledgements
Not applicable

Authors' contributions
H.L. conducted the experiments, analyzed the data and wrote the paper; X.L. proposed the idea and revised the paper; H.L. and X.L. designed the experiments; L.J., R.L., J.D. and J.Y. reviewed the experiments and paper. All authors have read and agreed to the published version of the manuscript.

Authors' information
Name: Xiaowu Li
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
Name: Haochen Lv
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
Name: Lianyin Jia
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
Name: Runxin Li
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
Name: Jiaman Ding
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China
Name: Jinguo You
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650504, China

Funding
The research is supported by Grants from the National Natural Science Foundation of China (No. 62062046, 61562054, 51467007, 61462050), Training Programme Foundation for the Talents by Kunming university of science and technology (Grant No. KKSY201603016), and the Scientific Research Program Foundation of Hubei University of Technology (Grant No. BSQD121270).

Availability of data and materials
All data, models, and code generated or used during the study appear in the submitted article.

Competing interests
The authors declare no conflict of interest.

Author details
[1]Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China. [2]Key Lab of Computer Technologies Application of Yunnan Province, Kunming 650504, China. [3]Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650504, China.

Received: 17 September 2020 Accepted: 14 January 2021
Published online: 29 January 2021

References

1. Tewari A, Gupta B (2017) Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J Supercomput 73: 1085–1102
2. Hossain K, Rahman M, Roy S (2019) Iot data compression and optimization techniques in cloud storage: current prospects and future directions. Int J Cloud Appl Comput (IJCAC) 9:43–59
3. Adat V, Gupta B (2018) Security in internet of things: issues, challenges, taxonomy, and architecture. Telecommun Syst 67:423–441
4. Aslam B, Kashif M, Awais Azam M, Amin Y, Loo J, Tenhunen H (2019) A low profile miniature RFID tag antenna dedicated to IoT applications. Electromagnetics 39:393–406
5. Han J, Qian C, Wang X, Ma D, Zhao J, Xi W, Jiang Z, Wang Z (2015) Twins: device-free object tracking using passive tags. IEEE/ACM Trans Networking 24:1605–1617
6. Yan L, Xiong D (2018) Mobile motion robot indoor passive RFID location research. Int J RF Technol 9:113–129
7. Amato F, Torun HM, Durgin GD (2018) RFID backscattering in long-range scenarios. IEEE Trans Wirel Commun 17:2718–2725
8. Liu X, Xie X, Wang S, Liu J, Yao D, Cao J, Li K (2019) Efficient range queries for large-scale sensor-augmented RFID systems. IEEE/ACM Trans Networking 27: 1873–1886
9. Zhou WH, De Jiang N, Yan CC (2019) Research on anti-collision algorithm of RFID tags in logistics system. Proc Comput Sci 154:460–467
10. Biswal AK, Jenamani M, Kumar SK (2018) Warehouse efficiency improvement using RFID in a humanitarian supply chain: implications for Indian food security system. Transportation Res Part E Logistics Transportation Rev 109:205–224
11. Zhang L, Xiang W, Tang X (2017) An efficient bit-detecting protocol for continuous tag recognition in mobile RFID systems. IEEE Trans Mob Comput 17:503–516
12. Satyavolu CJ, Radhakrishnan S, Sarangan V, Landers TL, Veeramani M (2016) Mobile RFID tag reading with non-overlapping tandem readers on a conveyor belt. Ad Hoc Netw 45:22–33
13. Jo M, Youn HY, Cha S-H, Choo H (2009) Mobile RFID tag detection influence factors and prediction of tag detectability. IEEE Sensors J 9:112–119
14. Yeh MK, Jiang JR, Huang ST (2014) Four–ary query tree splitting with parallel responses for RFID tag anti–collision. Int J Ad Hoc Ubiquitous Comput 16:193–205
15. Landaluce H, Perallos A, Onieva E, Arjona L, Bengtsson L (2016) An energy and identification time decreasing procedure for memoryless RFID tag anticollision protocols. IEEE Trans Wirel Commun 15:4234–4247
16. Knerr B, Holzer M, Angerer C, Rupp M (2010) Slot-wise maximum likelihood estimation of the tag population size in FSA protocols. IEEE Trans Commun 58:578–585
17. Vales-Alonso J, Bueno-Delgado V, Egea-Lopez E, Gonzalez-Castano FJ, Alcaraz J (2011) Multiframe maximum-likelihood tag estimation for RFID anticollision protocols. IEEE Trans Ind Inform 7:487–496
18. Lin X-H, Wang H, Kwok Y-K, Chen B, Dai M, Zhang L (2015) Exploiting the prefix information to enhance the performance of FSA-based RFID systems. Comput Commun 56:108–118
19. Solic P, Blazevic Z, Skiljo M, Patrono L, Colella R, Rodrigues JJ (2017) Gen2 RFID as IoT enabler: characterization and performance improvement. IEEE Wirel Commun 24:33–39
20. Wong C, Feng Q (2007) Grouping based bit-slot ALOHA protocol for tag anti-collision in RFID systems. IEEE Commun Lett 11:946–948
21. Chen W-T (2008) An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA. IEEE Trans Autom Sci Eng 6:9–15
22. HajMirzaei M (2019) Novel tag estimation method by use of Manchester coding in RFID systems. Int J Commun Syst 32:e4101
23. Wang Z, Huang S, Fan L, Zhang T, Wang L, Wang Y (2018) Adaptive and dynamic RFID tag anti-collision based on secant iteration. PLoS One 13: e0206741
24. Jiang Z, Li B, Yang M, Yan Z (2020) LC-DFSA: low complexity dynamic frame slotted Aloha anti-collision algorithm for RFID system. Sensors 20:228
25. Li J, Pang X (2018) Belt conveyor dynamic characteristics and influential factors. Shock Vib 2018:1–13
26. He D, Pang Y, Lodewijks G (2017) Green operations of belt conveyors by means of speed control. Appl Energy 188:330–341
27. Saxl G, Ferdik M, Fischer M, Maderboeck M, Ussmueller T (2019) UHF RFID prototyping platform for ISO 29167 decryption based on an SDR. Sensors 19:2220
28. He C, Wang ZJ, Miao C, Leung VC (2015) Block-level unitary query: enabling orthogonal-like space-time code with query diversity for MIMO backscatter RFID. IEEE Trans Wirel Commun 15:1937–1949
29. Ko H, Mesicek L, Choi J, Choi J, Hwang S (2018) A study on secure contents strategies for applications with drm on cloud computing. Int J Cloud Appl Comput (IJCAC) 8:143–153
30. Stergiou C, Psannis KE, Gupta BB, Ishibashi Y (2018) Security, privacy & efficiency of sustainable cloud computing for big data & IoT. Sustainable Comput Inform Syst 19:174–184
31. Li X, Feng Q (2013) Grouping based dynamic framed slotted ALOHA for tag anti-collision protocol in the mobile RFID systems. Appl Math Inf Sci 7:655–660

Publisher's Note