

Dynamic Framed-Slot ALOHA Anti-Collision using Precise Tag Estimation Scheme

Author

Pupunwiwat, Prapassara, Stantic, Bela

Published

2010

Conference Title

Twenty-First Australasian Database Conference (ADC2010)

Rights statement

© 2010 Australian Computer Society Inc. The attached file is posted here in accordance with the copyright policy of the publisher, for your personal use only. No further distribution permitted. Use hypertext link for access to the conference website.

Downloaded from

<http://hdl.handle.net/10072/31360>

Link to published version

<http://dl.acs.org.au/>

Griffith Research Online

<https://research-repository.griffith.edu.au>

Dynamic Framed-Slot ALOHA Anti-Collision using Precise Tag Estimation Scheme

Prapassara Pupunwiwat

Bela Stantic

Institute for Integrated and Intelligent Systems
Griffith University, Queensland, Australia
Email: {p.pupunwiwat, b.stantic}@griffith.edu.au

Abstract

Radio Frequency Identification (RFID) technology uses radio-frequency waves to automatically identify people or objects. Despite an emergence of RFID technology, multiple tag identification, where a reader identifies a multiple number of tags in a very short time, is still a major problem. This is known as *Collision* problem and can be solved by using anti-collision scheme. The current tree-based anti-collision approach suffers from long identification delay, while the ALOHA-Based approach suffers from tag starvation problem due to inaccurate Frame-size. In this paper, we propose a “Precise Tag Estimation Scheme” for a *Dynamic Framed-Slot ALOHA* (DFSA), which estimates precise number of tags around the reader. In this empirical study, we compare our approaches with the original tag estimation in DFSA. The results indicate that the various parameters used by “Precise Tag Estimation Scheme”, including *empty slots* variables and/or *collision slots* variables, have an impact on system efficiency. Thus, the number of frames and slots used by DFSA can be minimised by adjusting correct variables.

Keywords: Radio Frequency Identification - RFID, Anti-Collision, Frame-Size Estimation, Backlog Estimation

1 Introduction

RFID technology has gained significant momentum in the past few years. It has promised to improve the efficiency of business processes by providing the automatic identification and data capture. The core RFID technology is not new, and it can be traced back to World War II where it was used to distinguish between friendly and enemy aircrafts or known as friend-or-foe (Landt 2001). Currently RFID technology is used in different systems such as: transportation, distribution, retail and consumer packaging, security and access control, monitoring and sensing, library system, defence and military, health care, and baggage and passenger tracing at the airports.

In a RFID system, when numerous tags are present in the interrogation zone at the same time, the reader requires an ability to read data from the individual tag. A technical scheme that handles tag collision without any interference is called an anti-collision protocol. A RFID reader is a powerful device that has sufficient power and memory, while a passive tag

requires energy from the radio signal sent by a reader. The reader issues a request command to the tags and each tag will send its ID to the reader. If only one tag responds, the reader can receive the tag's ID. However, if more than two tags send their IDs simultaneously, a collision occurs and the IDs need to be re-transmitted according to an anti-collision scheme. The main focus of an anti-collision scheme is to read multiple tags as fast and reliably as possible.

The two types of tag anti-collision algorithms widely used in RFID systems are the *Tree-based Deterministic Anti-collision* and the *ALOHA-based Probabilistic Anti-collision*. Tree-based algorithms make trees while performing the tag identification procedure using a unique ID of each tag, which lead to lengthy queries issued by reader that causes long identification delay. On the other hand, ALOHA-based algorithms decreases the probability of collision by scheduling the responses of tags at random time, which lead to tag starvation problems where not all tags can be identified.

In this study, we propose a new “Precise Tag Estimation Scheme” (PTES) for Backlog estimation and Frame-size estimation compatible with Dynamic Framed-Slot ALOHA. The motivation of this work is to achieve a more accurate estimation of number of tags within an interrogation zone, which leads to a more accurate frame-size and system efficiency. The methodology for PTES is first derived, and experiment is then conducted in order to prove the efficiency of the proposed technique. The results and analysis of the experiments have indicated that PTES, using various parameters, has an impact on DFSA system efficiency and the number of frames and slots used can be minimised by adjusting the correct variables.

The remainder of this paper is organised as follows: In section 2, some general background on RFID and information related to collision and different anti-collision schemes are provided. In section 3, we discuss different probabilistic anti-collision schemes compatible with EPC Gen2 and their limitations. In section 4, we present a new technique, the “Precise Tag Estimation Scheme” and methodology. In section 5, we present experimental evaluation, results, analysis and discussions, and finally in section 6 we provide our conclusion and future work.

2 RFID Background

RFID may only consist of a tag and a reader but a complete RFID system involves many other components, such as computer, network, Internet, and software such as middleware and user applications. A typical RFID system is divided into two layers: the physical layer and Information Technology (IT) layer (Brown et al. 2007).

2.1 Tag and Reader Collision

Simultaneous transmissions in RFID systems lead to collisions as the readers and tags typically operate on the same channel. Three types of collisions are possible: Reader-Reader collision, Reader-Tag collision, and Tag-Tag collision.

- **Reader-to-Reader:** Interference occurs when one reader transmits a signal that interferes with the operation of another reader and prevents the second reader from communicating with tags in its interrogation zone (Jain & Das 2006). Reader-to-Reader collision can be easily avoided by determining the appropriate reader’s deployment that prevents direct signal interference between two or more readers.
- **Reader-to-Tag:** Interference occurs when one tag is simultaneously located in the interrogation zone of two or more readers, where more than one reader attempts to communicate with that tag at the same time (Jain & Das 2006).
- **Tag-to-Tag:** Tag collision in RFID systems, also known as Multi-access, happens when multiple tags are energised by the RFID tag reader simultaneously, and reflect their respective signals back to the reader at the same time. This problem is often seen whenever a large volume of tags must be read together in the same reader zone. The reader is unable to differentiate these signals.

2.2 Tag Anti-Collision Approaches

Tag collision or Multi-access problem is more complex than those within reader collision categories. Therefore, in this paper we only focus on tag anti-collision approaches. The various types of tag anti-collision approaches for Multi-access/Tag collision can be reduced to two basic types: Tree-based approach and ALOHA-based approach.

2.2.1 Tree-Based Approaches

The Tree-based approach starts by asking for the first number of the tag (Query Tree algorithm) until it matches the tags; then it continues to ask for additional characters until all tags within the region are found. This approach is slow and introduces a long identification delay but leads to fewer collisions, and has 100 percent successful identification rate (Choi et al. 2008).

Such Tree-based approaches can be classified into a Memory-based algorithm and a Memoryless-based algorithm (Myung & Lee 2006a), (Myung & Lee 2006b), (Ryu et al. 2007). In the Memory-based algorithm, the reader’s inquiries and the responses of the tags are stored and managed in the tag memory, resulting in an equipment cost increase especially for RFID tags. In contrast, in the Memoryless-based algorithm, the responses of the tags are not determined by the reader’s previous inquiries. The tags’ responses and the reader’s present inquiries are determined only by the present reader’s inquiries so that the cost for the tags can be minimised.

2.2.2 ALOHA-Based Approaches

In an ALOHA-based approach, tags respond at randomly generated times. If a collision occurs, colliding tags will have to identify themselves again after waiting a random period of time. This technique is faster than Tree-based but suffers from tag starvation problem where not all tags can be identified due to the random nature of chosen time.

The ALOHA-based approaches usually refer to “Slot ALOHA” (Quan et al. 2006), which introduces discrete time-slots for tags to be identified by reader at the specific time. The principle of Slotted-ALOHA techniques is based on the “Pure ALOHA” introduced in early 1970s (Abramson 1970), where each tag is identified randomly. To improve the performance and throughput rate, a “Frame Slotted ALOHA” (Shin et al. 2007) anti-collision algorithm has been suggested, where each frame is formed of specific number of slots that is used for the communication between the readers and the tags. Each tag in the interrogation zone arbitrarily selects a slot for transmitting the tag’s information. However, the Frame Slotted ALOHA uses a fixed frame-size and does not change the frame-size during the process of tag identification. This is simple, but not efficient for tag identification. The “Dynamic Framed Slotted ALOHA” algorithm can change the frame-size to increase the efficiency of tag identification; and there have been several researches to improve the accuracy of frame-size by implementing a Frame Estimation Tool (Wang et al. 2007), (Lee et al. 2005), (Lee et al. 2008), (Cho et al. 2007).

In this study, we focus on improving the Frame-size estimation, since EPC Class 1 Gen2 protocol (EPCGlobal 2006) adopted Dynamic Framed-Slot ALOHA probabilistic algorithm to solve the collision problem.

3 Dynamic Framed-Slot ALOHA Algorithm

Probabilistic algorithms are based on ALOHA protocol. Each tag in an interrogation zone selects one of the given N slots to transmit its identifier. All tags will be recognised after a few frames. Figure 1 shows an example of Dynamic Framed-Slot ALOHA anti-collision protocols. Each frame is formed of specific number of slots that is used for the communication between the readers and the tags. The frame-size is dynamically changed according to estimated number of ‘Backlog’, which is a number of tags that have not been read. Any slot that has more than one tags responding to it is classified as a collision slot, while any slot that has exactly one tag responding to it is a successful slot. In addition, empty slot is where there is no tag reply and should be minimised in order to increase system efficiency. Figure 1 shows that Slot 1 and 2 of Frame one and Slot 5 of Frame two are collision slots; and Slot 3 of Frame one, Slot 4 of Frame two, and Slot 6 and 7 of Frame three are successful slots.

Frame	1			2		3	
Slot	1	2	3	4	5	6	7
State	Collision	Collision	Success	Success	Collision	Success	Success
Tag A							
Tag B							
Tag C							
Tag D							
Tag E							

Figure 1: This figure shows a sample procedure of Dynamic Framed-Slot ALOHA.

3.1 DFSA for EPC Class 1 Gen2

EPC Class 1 Generation 2 is widely used in Ultra High Frequency (UHF) range for communications at 860-960MHz; where the standards have been created by EPCGlobal (EPCGlobal 2006). EPC Class 1 Gen2

protocol adopted the Dynamic Framed-Slot ALOHA-based probabilistic algorithm to solve the collision problem.

According to the protocol, the reader picks tag within an interrogation zone by the command “SELECT”; then issues “QUERY”, which contains a ‘Q’ parameter to specify the frame-size (frame-size $F = 2^Q - 1$). Each selected tag will pick a random number between 0 to $2^Q - 1$ and put it into its slot counter. The tag, which picks zero as its slot number, will respond to the reader with a ‘RN16’. After receiving the RN16 from tag, reader will transmit ‘ACK’ containing the received RN16, after which the tag with slot number zero will backscatter its EPC to reader. Then, reader issues “QUERYREP” or “QUERYADJUST” command to initiate another slot (Wang et al. 2007).

In order to query the remaining tags, reader may issue “QUERYREP” and each tag will subtract 1 from its own slot number, or “QUERYADJUST” (Adjust Q value); and each tag will pick a new random number within 0 to $2^Q - 1$ as its new slot number. The tag whose new slot number is zero, will response to reader and then backscatter its EPC (Li et al. 2009). There are three kinds of slot: 1) Successful slot where there is only one tag reply, 2) Empty slot where there is no tag reply, and 3) Collision slot where there is more than one tag reply; as shown in figure 2.

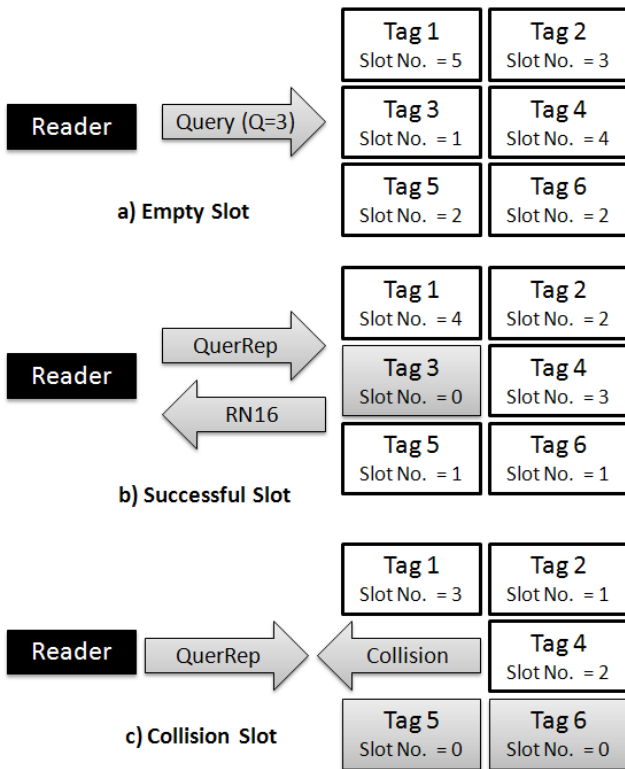


Figure 2: Empty Slot, Successful Slot, and Collision Slot in EPC Class 1 Gen2 Protocol.

3.2 Mathematic Fundamental for ALOHA-based Tag Estimation

In the ALOHA-based probabilistic scheme, to estimate the number of present tags, Binomial distribution is a good fundamental method (Wang et al. 2007), (Li et al. 2009). For a given initial Q in a frame with F slots and n tags, the expected value of the number of slots with occupancy number x is as follows:

$$a_x = n \times C_n^x \left(\frac{1}{F}\right)^x \left(1 - \frac{1}{F}\right)^{n-x}$$

Therefore, the expected number of Empty slot e , Successful slot s , and Collision slot c are given by the following equations:

$$\begin{cases} e = a_0 = F \left(1 - \frac{1}{F}\right)^n \\ s = a_1 = n \left(1 - \frac{1}{F}\right)^{n-1} \\ c = a_k = F - a_0 - a_1 \end{cases}$$

Thus, the system efficiency is defined as the ratio between Successful slot number and Frame-size given by the following equations:

$$E = \frac{s}{F} = \frac{n \left(1 - \frac{1}{F}\right)^{n-1}}{F} = n \frac{1}{F} \left(1 - \frac{1}{F}\right)^{n-1}$$

It is proven that the highest efficiency can be obtained if the Frame-size is equal to the number of tags, provided that all slots have the same fixed length:

$$F(\text{optimal}) = n$$

3.2.1 Schoute Backlog Estimation Technique

Schoute (1983) developed a Backlog estimation technique for Dynamic Framed-Slot ALOHA using Poisson distribution. The Backlog, after the current frame Bt, is given by equation:

$$Bt = 2.39c$$

where c represents the number of collided slot in the current frame. This technique has the best performance, where fewest frames were used compared to other algorithms.

3.2.2 Lowerbound Backlog Estimation Technique

The estimation function is obtained under the assumption that a collision involves at least two different tags. Therefore, Backlog after the current frame Bt is given by equation:

$$Bt = 2c$$

where c is the number of collided slot in the current frame.

3.2.3 Other Backlog Estimation Techniques

There have been several researches on Backlog estimation including C-Ratio method (Cha & Kim 2005), Chen1 and Chen2 methods (Chen 2006), Vogt method (Vogt 2002), and Bayesian method (Floerke-meier 2007). These methods are either having worse performances than simple Schoute’s method or too complicated to implement for RFID system. Therefore, we only compare our method to Schoute and Lowerbound methods because the two methods are simple and have excellent performances.

4 Methodology

In order to overcome shortcomings of inaccurate Frame-size estimation methods, we propose a “Precise Tag Estimation Scheme” (PTES), which introduces three tag estimation methods compatible with *Dynamic Framed-Slot ALOHA* used in EPC Class

1 Generation 2. Each method of the PTES estimates specific number of tags according to a specific equation. After precise number of tags is estimated, frame-size for the next round of identification process can be predicted. This section will describe the specific requirements for tag estimation, initial Q value, suggest frame-size, the newly proposed PTES, and sample tag estimation and allocation.

4.1 Slots Observation and Initial Q-Value

According to DFSA algorithm, the reader picks tag within an interrogation zone by the command “SELECT”; then issues “QUERY”, which contains a ‘Q’ parameter to specify the frame-size, $F = 2^Q - 1$. For our methodology, initial Q value can be any number between 1 and 10 since we assume that a reader can at most pick up no more than 800 tags per round. After first round of identification, collision slots and empty slots will be observed and used to estimate number of tags. Three methods are proposed for Precise Tag Estimation Scheme: 1) Method One - Various Empty Parameters, 2) Method Two - Various Collision Parameters, and 3) Method Three - Various Collision and Empty Parameters. The three methods consist of different equations for number of tags estimation. After the number of tags has been estimated, frame-size for the next identification round can be configured. The suggested frame-size is explained in the following sub-section.

4.2 Suggested Frame-Size

The suggested frame-size for our methodology is set according to the estimated number of tags. For example, if an estimated number of tags are to be around 100 tags, the suggested frame-size would have a Q value of 7. Since the frame-size is calculated by $2^Q - 1$, the frame-size where $Q = 7$ will allow at most 128 tags (0 to $2^7 - 1$) to be identified. Therefore, if the estimated number of tags is between 65 and 128 tags, the suggested Q would equal to 7. Table 1 shows minimum and maximum number of tags allowed per suggested frame-size. The table only demonstrates up to $Q = 10$ since we assume that no more than 800 tags will be captured each round by the reader.

2^Q	2^1	2^2	2^3	2^4	2^5
Min	0	3	5	9	17
Max	2	4	8	16	32
2^Q	2^6	2^7	2^8	2^9	2^{10}
Min	33	65	129	257	513
Max	64	128	256	512	1024

Table 1: Suggested Frame-size for specific estimated number of tags. Maximum tags of each Frame-size is calculated by $2^Q - 1$

4.3 Precise Tag Estimation Scheme

In this paper, we propose three tag estimation methods for a Precise Tags Estimation Scheme. In method One - *Various Empty Parameters* (VEP), we use a fix parameter to calculate collision slot and use a variable to predict empty slot for the next round of identification. In method two - *Various Collision Parameters* (VCP), we use a variable to predict collision slot for the next round. Finally, in method three - *Various Collision and Empty Parameters* (VCEP), we use two variables to predict collision slot and empty slot for the new identification round. The aims of all methods

are to clarify that both collision slots and empty slots have an impact on backlog prediction; and that more than one variable can be used to predict Frame-size for an upcoming round effectively. The three methods are explained within this sub-section.

4.3.1 Method One - VEP

Various Empty Parameters method intends to obtain the optimal parameter in order to calculate and predict the closest number of remaining tags for the next identification round. We assume that for the current identification round, each collision slot has at least 2 tags collided. However, we cannot know for sure how many tags actually caused the collision. There is exactly 1 tag per successful slot, thus, we do not take successful slot into consideration. On the other hand, empty slot will continuously occur during the next rounds of identification. Therefore, VEP method is created to find the optimal parameter to predict the number of empty slot; and to find the impact of variable used on empty slots prediction.

The VEP method uses a variable between 0.1 and 0.9 to predict the number of empty slot. Since empty slot does not engage any tag, we assume that the parameter for empty slot calculation falls between 0.1 and 0.9 (more than 0 but less than 1). Equation (1) shows *Backlog* (Number of unidentified tags) estimation using fix parameter $V_1 = 2.0$ for collision slot prediction and variable $0 < V_2 < 1$ for empty slot prediction.

$$Backlog = V_1(c) + V_2(e).....(1)$$

where c = Collision Slot; e = Empty slot; $V_1 = 2.0$; and $V_2 =$ Variable between 0.1 and 0.9 incrementing by 0.1.

$$Backlog = 2.0(c) + 0.1(e)$$

$$Backlog = 2.0(c) + 0.2(e)$$

.....

$$Backlog = 2.0(c) + 0.9(e)$$

Therefore, there are 9 possible optimal V_2 for this method.

4.3.2 Method Two - VCP

Various Collision Parameters method intends to obtain the optimal parameter in order to calculate and predict the closest number of remaining tags for the next identification round. Similarly to VEP method, we assume that for the current identification round, each collision slot has at least 2 tags collided. However, we cannot know for sure how many tags actually caused the collision. There is exactly 1 tag per successful slot, therefore, we do not take successful slot into consideration. In addition, empty slot does not engage any tag. Accordingly, we also do not take empty slot into consideration. VCP method focuses on finding optimal variable to calculate and predict the number of collision slot for the next identification round; and to find the impact of variable used on collision slots prediction.

The VCP method uses different parameter between 2.0 and 3.0 to predict the number of collision slot. Since collision slot engage at least 2 tags, we assume that the parameter for collision slot calculation falls between 2.0 and 3.0 (more than 2 but possibly less than 3). However, number of tags per collision slot can be more than 3 tags. According to Schoute’s method (Schoute 1983), which has the best Backlog estimation, the parameter used is 2.39. Therefore, we

assumed that the optimal parameter falls between 2 and 3. Equation (2) shows Backlog estimation using variable $2.0 \leq V_1 \leq 3.0$ for collision slot prediction.

$$Backlog = V_1(c) \dots \dots \dots (2)$$

where c = Collision Slot; and V_1 = Variable between 2.0 and 3.0 incrementing by 0.1.

$$Backlog = 2.0(c)$$

$$Backlog = 2.1(c)$$

.....

$$Backlog = 3.0(c)$$

Therefore, there are 11 possible optimal V_1 for this method.

4.3.3 Method Three - VCEP

Various Collision and Empty Parameters method intends to obtain the optimal parameter in order to calculate and predict the closest number of remaining tags for the next identification round. In this method, we also assume that for the current identification round, each collision slot has at least 2 tags collided, but we do not know how many tags actually caused the collision. There is exactly 1 tag per successful slot, thus, we do not take successful slot into consideration. On the other hand, empty slot will continuously occur during the next rounds of identification. Therefore, VCEP method is created to find the optimal parameters to predict the number of both collision slot and empty slot for the upcoming round.

The VCEP method uses variable between 2.0 and 3.0 to predict the number of collision slot. Since collision slot engage at least 2 tags, we assume that the parameter for collision slot calculation falls between 2.0 and 3.0. Variable between 0.1 and 0.9 is also used to predict the number of empty slot. Since empty slot does not engage any tag, we assume that the parameter for empty slot calculation falls between 0.1 and 0.9. Equation (3) shows Backlog estimation using variable $2.0 \leq V_1 \leq 3.0$ for collision slot prediction and variable $0 < V_2 < 1$ for empty slot prediction.

$$Backlog = V_1(c) + V_2(e) \dots \dots \dots (3)$$

where c = Collision Slot; e = Empty slot; V_1 = Variable between 2.0 and 3.0; and V_2 = Variable between 0.1 and 0.9 incrementing by 0.1.

$$Backlog = 2.0(c) + 0.1(e)$$

$$Backlog = 2.0(c) + 0.2(e)$$

.....

$$Backlog = 2.0(c) + 0.9(e)$$

$$Backlog = 2.1(c) + 0.1(e)$$

.....

.....

$$Backlog = 3.0(c) + 0.8(e)$$

$$Backlog = 3.0(c) + 0.9(e)$$

Therefore, there are 99 possible optimal V_1V_2 for this method. Note that some parameters from VEP method are also included in this method.

Figure 3 shows 99 possible optimal V_1V_2 variables for c and e . The optimal V_1V_2 for this method will be different from optimal variable of VEP and VCP since V_1 and V_2 of VCEP have impact on each other.

4.4 Sample Tag Allocation and Estimation

We are now initiating a sample tag allocation and estimation. Figure 4 shows a sample of first round tag allocation, where six collision slots, four empty slots, and six successful slots occurred. For each collision slot, two or more tags collided; while empty slot engaged no tag. Each successful slot holds exactly one tag per slot. After the first round of tag allocation, VEP, VCP, and VCEP equations are applied in order to find an estimated frame-size for the next identification round.

Slot type	s	e	s	c	c	e	s	e	c	s	c	e	c	s	c	s
Tag per slot	1	0	1	2	2	0	1	0	2	1	2	0	2	1	2	1

Figure 4: A sample First Round of Tag Allocation with Initial Q of 4. Collision slot $c = 6$, Empty slot $e = 4$, and Successful slot $s = 6$.

4.4.1 Sample Estimation - VEP

After the first round of identification shown in figure 4, VEP method uses variable V_2 between 0.1 to 0.9 to estimate empty slot and fixed V_1 of 2.0 to estimate collision slot for the next round. After applying Equation (1), number of estimated tags for the next round are as shown in table 2 under round one ($c = 6$, $e = 4$). We can see that by using various parameters of V_2 , different numbers of tags were predicted. For example, where $V_2 = 0.2$, the number of estimated tag can be calculated as follows:

$$Backlog = 2.0(6) + 0.2(4) = 12.8 \approx 13$$

Therefore, the estimated number of tags for the next round is equal to 13 tags. Hence, the new Q adjust is equal to 4 (see table 1).

Round one (c = 6, e = 4)		
Variable (V_2)	Tag Estimation	Q Adjust
0.1	12	4
0.2	13	4
0.3	13	4
0.4	14	4
0.5	14	4
0.6	14	4
0.7	15	4
0.8	15	4
0.9	16	4
Round two (c = 3, e = 6)		
Variable (V_2)	Tag Estimation	Q Adjust
0.1	7	3
0.2	7	3
0.3	8	3
0.4	8	4
0.5	9	4
0.6	10	4
0.7	10	4
0.8	11	4
0.9	11	4

Table 2: Sample tag estimation and Frame-size (Q) adjustment using Method One - VEP.

Accordingly, after the first round, the adjustment of Q values using different parameters for tag estimation is equal to 4. In order to further demonstrate the estimation of frame-size, a sample of second round tag allocation is shown in figure 5.

2.0c, 0.1e	2.0c, 0.2e	2.0c, 0.3e	2.0c, 0.4e	2.0c, 0.5e	2.0c, 0.6e	2.0c, 0.7e	2.0c, 0.8e	2.0c, 0.9e
2.1c, 0.1e	2.1c, 0.2e	2.1c, 0.3e	2.1c, 0.4e	2.1c, 0.5e	2.1c, 0.6e	2.1c, 0.7e	2.1c, 0.8e	2.1c, 0.9e
2.2c, 0.1e	2.2c, 0.2e	2.2c, 0.3e	2.2c, 0.4e	2.2c, 0.5e	2.2c, 0.6e	2.2c, 0.7e	2.2c, 0.8e	2.2c, 0.9e
2.3c, 0.1e	2.3c, 0.2e	2.3c, 0.3e	2.3c, 0.4e	2.3c, 0.5e	2.3c, 0.6e	2.3c, 0.7e	2.3c, 0.8e	2.3c, 0.9e
2.4c, 0.1e	2.4c, 0.2e	2.4c, 0.3e	2.4c, 0.4e	2.4c, 0.5e	2.4c, 0.6e	2.4c, 0.7e	2.4c, 0.8e	2.4c, 0.9e
2.5c, 0.1e	2.5c, 0.2e	2.5c, 0.3e	2.5c, 0.4e	2.5c, 0.5e	2.5c, 0.6e	2.5c, 0.7e	2.5c, 0.8e	2.5c, 0.9e
2.6c, 0.1e	2.6c, 0.2e	2.6c, 0.3e	2.6c, 0.4e	2.6c, 0.5e	2.6c, 0.6e	2.6c, 0.7e	2.6c, 0.8e	2.6c, 0.9e
2.7c, 0.1e	2.7c, 0.2e	2.7c, 0.3e	2.7c, 0.4e	2.7c, 0.5e	2.7c, 0.6e	2.7c, 0.7e	2.7c, 0.8e	2.7c, 0.9e
2.8c, 0.1e	2.8c, 0.2e	2.8c, 0.3e	2.8c, 0.4e	2.8c, 0.5e	2.8c, 0.6e	2.8c, 0.7e	2.8c, 0.8e	2.8c, 0.9e
2.0c, 0.1e	2.9c, 0.2e	2.9c, 0.3e	2.9c, 0.4e	2.9c, 0.5e	2.9c, 0.6e	2.9c, 0.7e	2.9c, 0.8e	2.9c, 0.9e
3.0c, 0.1e	3.0c, 0.2e	3.0c, 0.3e	3.0c, 0.4e	3.0c, 0.5e	3.0c, 0.6e	3.0c, 0.7e	3.0c, 0.8e	3.0c, 0.9e

Figure 3: Variable V_1 and V_2 for Collision slot and Empty slot calculation in method three - VCEP. There are 99 possible combinations of V_1 and V_2 in order to find optimal parameters for c and e prediction.

Slot type	s	s	e	e	e	s	e	e	e	s	c	c	s	s	s	c
Tag per slot	1	1	0	0	0	1	0	0	0	1	2>	2>	1	1	1	2>

Figure 5: A sample second round of tag allocation where $Q = 4$. Collision slot $c = 3$, Empty slot $e = 6$, and Successful slot $s = 7$.

After the second round of identification shown in figure 5, VEP method again uses variable V_2 between 0.1 to 0.9 to estimate empty slot and fixed V_1 of 2.0, to estimate collision slot for the next round. After applying Equation (1), numbers of estimated tags for the next round are as shown in table 2 under round two ($c = 3$, $e = 6$). We can see that different numbers of tags were predicted and parameter Q is adjusted according to specific number of tags. For example, where $V_2 = 0.7$, the number of estimated tag can be calculated as follows:

$$\text{Backlog} = 2.0(3) + 0.7(6) = 10.2 \approx 10$$

Therefore, the estimated number of tags for the next round is equal to 10 tags. Hence, the new Q adjust is equal to 4 (see table 1).

Following the second round, the adjustment of Q value for parameters $V_2 = 0.1 - 0.3$ is equal to 3, while the Q value for $V_2 = 0.4 - 0.9$ is equal to 4. In order to identify all tags within the interrogation zone, VEP with fixed parameter V_1 and variable V_2 is applied for each identification round until no more collision occur and all tags are identified.

4.4.2 Sample Estimation - VCP

After the first round of identification shown in figure 4, VCP method uses variable V_1 between 2.0 to 3.0 to estimate collision slot. By applying Equation (2), numbers of estimated tags for the next round are shown in table 3. By using various parameters of V_1 , different numbers of tags were predicted. For example, where $V_1 = 2.8$, the number of estimated tags can be calculated as follows:

$$\text{Backlog} = 2.8(6) = 16.8 \approx 17$$

Therefore, the estimated number of tags for the next round is equal to 17 tags. Hence, the new Q adjust is equal to 5 (see table 1).

Subsequently to the first round of identification, the adjustment of Q value for parameters $V_1 = 2.0 - 2.7$ is equal to 4, while the Q value for $V_1 = 2.8 - 3.0$ is equal to 5. In order to identify all tags within the interrogation zone, VCP with variable V_1 is applied

for each identification round until no more collision occurs and all tags are identified.

Round one (c = 6)		
Variable (V_1)	Tag Estimation	Q Adjust
2.0	12	4
2.1	13	4
2.2	13	4
2.3	14	4
2.4	14	4
2.5	15	4
2.6	16	4
2.7	16	4
2.8	17	5
2.9	17	5
3.0	18	5

Table 3: Sample tag estimation and Frame-size (Q) adjustment using method two - VCP.

4.4.3 Sample Estimation - VCEP

Following the first round of identification shown in figure 4, VCEP method uses variable V_1 between 2.0 to 3.0 to estimate collision slot and variable V_2 between 0.1 to 0.9, to estimate empty slot for the next round. Also, after applying Equation (3), numbers of estimated tags for the next round are shown in table 4. Different numbers of tags are predicted depending on various parameters of V_1 and V_2 . For example, where $V_1 = 2.5$ and $V_2 = 0.5$, the number of estimated tag can be calculated as follows:

$$\text{Backlog} = 2.5(6) + 0.5(4) = 17$$

Therefore, the estimated number of tags for the next round is equal to 17 tags. Hence, the new Q adjust is equal to 5 (see table 1).

After the first round, the adjustment of Q value where estimated tag is between 12 and 16 is equal to 4, while the Q value where estimated tag is between 17 and 22 is equal to 5. Likewise with VEP and VCP methods, VCEP with variable V_1 and V_2 is applied for each identification round until no more collision occurs, in order to identify all tags within the interrogation zone.

5 Experimental Evaluation

To measure a performance of "Precise Tag Estimation Scheme", we compare our results with performances of Schoute and Lowerbound methods. Different sets

Round one (c = 6, e = 4)																		
V ₂	0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9	
V ₁	Tag	Q	Tag	Q	Tag	Q	Tag	Q	Tag	Q	Tag	Q	Tag	Q	Tag	Q	Tag	Q
2.0	12	4	13	4	13	4	14	4	14	4	14	4	15	4	15	4	16	4
2.1	13	4	13	4	14	4	14	4	15	4	15	4	15	4	16	4	16	4
2.2	14	4	14	4	14	4	15	4	15	4	16	4	16	4	16	4	17	5
2.3	14	4	15	4	15	4	15	4	16	4	16	4	17	5	17	5	17	5
2.4	15	4	15	4	16	4	16	4	16	4	17	5	17	5	18	5	18	5
2.5	15	4	16	4	16	4	17	5	17	5	17	5	18	5	18	5	19	5
2.6	16	4	16	4	17	5	17	5	18	5	18	5	18	5	19	5	19	5
2.7	17	5	17	5	17	5	18	5	18	5	19	5	19	5	19	5	20	5
2.8	17	5	18	5	18	5	18	5	19	5	19	5	20	5	20	5	20	5
2.9	18	5	18	5	19	5	19	5	19	5	20	5	20	5	21	5	21	5
3.0	18	5	19	5	19	5	20	5	20	5	20	5	21	5	21	5	22	5

Table 4: A sample tag estimation and Frame-size (Q) adjustment using Method Three - VCEP.

of tags are considered and all three proposed algorithms (VEP, VCP, and VCEP) are implemented and applied for tag estimation.

Firstly, the reader picks tags within an interrogation zone by the command “SELECT”; then issues “QUERY” with ‘Q’ parameter to specify the frame-size. We have chosen the initial Q to be equal to 8. In the first round of identification, each selected tag will pick a random number between 0 to $2^8 - 1$ and place it into its slot counter. The tag which picks zero as its slot number will respond to the reader. Secondly, the reader issues “QUERYREP” or “QUERYADJUST” command to initiate another slot. If the “QUERYADJUST” command is issued, one of the Precise Tag Estimation Scheme will be applied and the new Q (Frame-size) will be predicted for the next round. Finally, in order to identify all tags within the interrogation zone, either VCP, VCEP, or VCEP will be applied after each identification round until no more collision occurs and all tags are identified.

Specification: An Intel Core 2 CPU with 2.40GHz processor and 3GB RAM is used for testing. A Microsoft Window XP professional with Service Pack 3 is installed on the computer. Algorithms for tags simulation are implemented using Java JCreator.

5.1 Data Set

We conducted three experiments using different tag sets: 1) the performance comparison of VEP method versus Schoute method and Lowerbound method; 2) the performance comparison of VCP method versus Schoute method and Lowerbound method; and 3) the performance comparison of VCEP method versus Schoute method and Lowerbound method. While performing DFSA anti-collision algorithm using various Tag estimation methods, number of tags is supposedly unknown.

- **Experiment One:** The aim of experiment one is to find the impact of Empty slot on Backlog estimation.
- **Experiment Two:** The aim of experiment two is to find the impact of different parameters used to predict number of Collision slot.
- **Experiment Three:** The aim of experiment three is to find the optimal parameters for Backlog estimation and next Frame-size prediction.

Table 5 shows that in experiment one, two, and three, there are three methods applied on a tag set of 300 tags. We performed ten runs on the data set and present the average results. Numbers in bracket show number of variable used for that method.

300 tags		
Ex 1	Ex 2	Ex 3
VEP (9)	VCP (11)	VCEP (99)
Sch (1)	Sch (1)	Sch (1)
LB (1)	LB (1)	LB (1)

Table 5: Data set used in experiment one, two, and three. Sch = Schoute method and LB = Lowerbound method. There are total of 11 variables used in experiment one, 13 variables in experiment two, and 101 variables in experiment three.

5.2 Results

Results are compared using both ‘Frame’ and ‘Slot’. Since frame-size is dynamic in DFSA, each frame involves different number of slot allocation. Thus, the method that used fewer frames may have higher slot counters than the other methods. In this sub-section, we present and clarify all experimental results and provide analysis and discussions.

5.2.1 Experiment One Result

Based on the experiment, figure 6 shows an average of ten runs of: a) number of slots, and b) number of frames, compared between VEP method ($V_1 = 2.0$, variable $0 < V_2 < 1$) versus Schoute (Sch) and Lowerbound (LB) methods. We can see that when V_2 is between 0.1 and 0.2, VEP uses less total slots (collision, empty, and successful slot) than both Sch and LB. When V_2 is between 0.3, 0.4, and 0.5, total number of slots of VEP method is slightly higher than Sch and LB. However, the number of frames used by VEP is lower. When V_2 is higher than 0.5, number of slots increased rapidly.

From the result, we conclude that empty slots have impacted on the Backlog estimation and the next frame-size prediction. We can also summarise that by using fix parameter $V_1 = 2$ and lower number for variable V_2 , especially where V_2 is between 0.1 and 0.5, the VEP can predict the number of Backlog more accurately.

5.2.2 Experiment Two Result

Figure 7 shows an average of ten runs of: a) number of slots, and b) number of frames, compared between VCP method (variable $2.0 \leq V_1 \leq 3.0$) versus Sch and LB methods. VCP uses only one variable V_1 to predict collision slot, where no empty slot was taken into consideration. By using VCP method, the results using different V_1 are similar to each other, where only variable $V_1 = 2.9$ and 3.0 have high number of slots used.

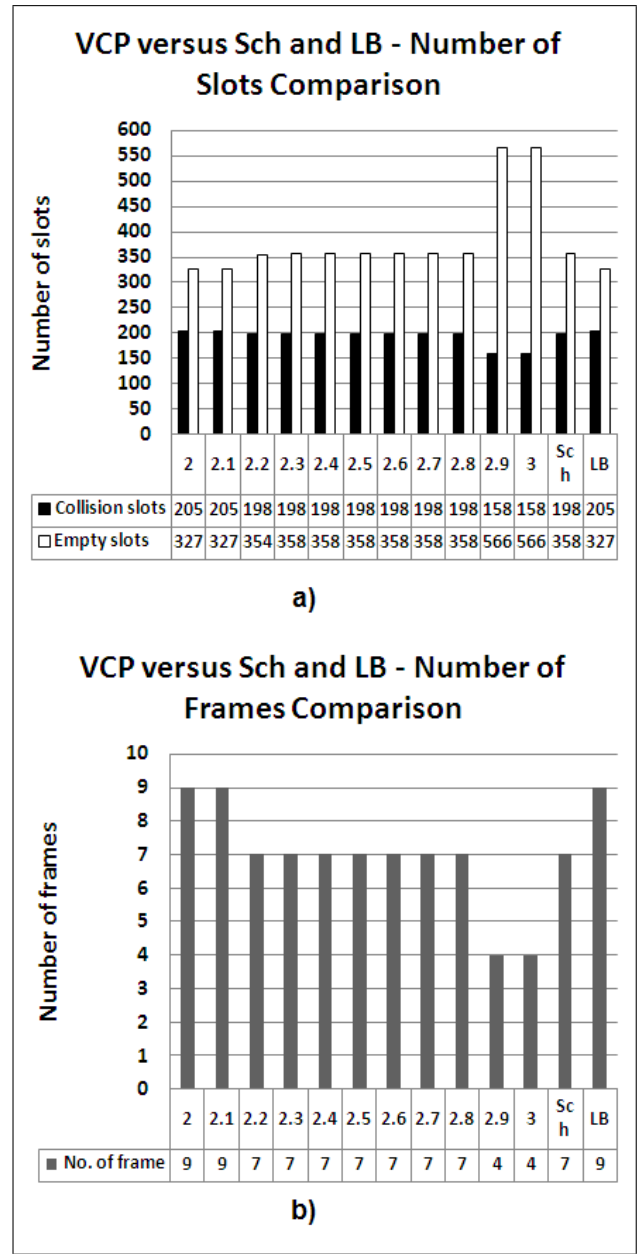
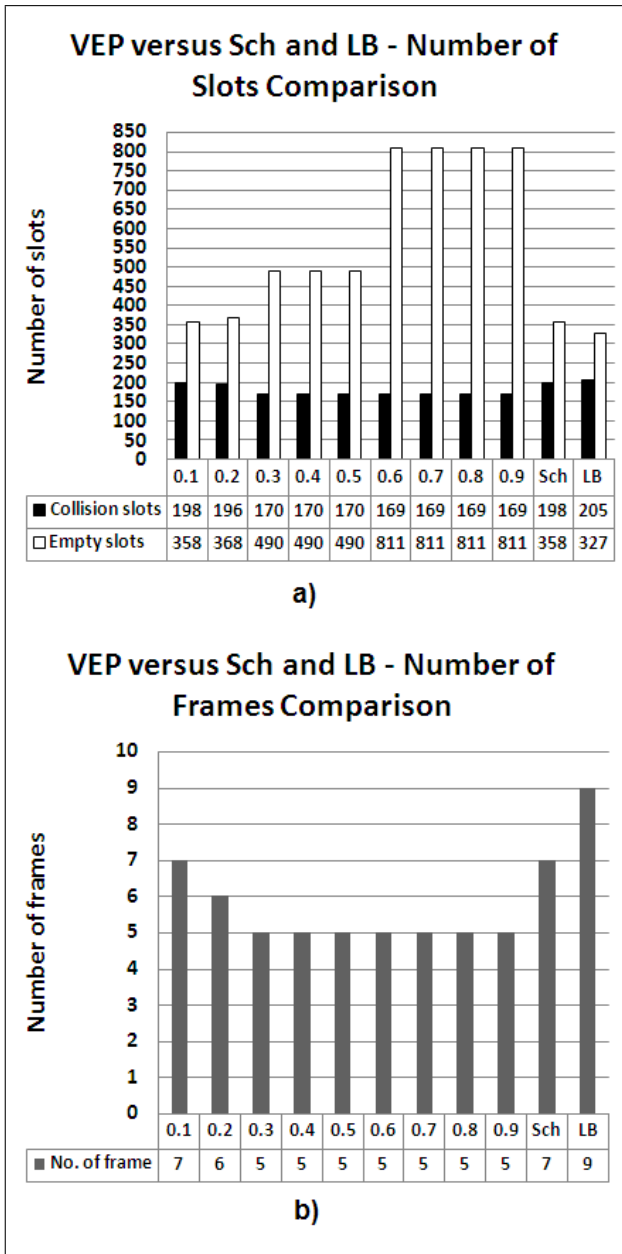


Figure 6: Comparison between VEP method with different variables versus Sch and LB methods: a) Number of slots, b) Number of frames.

From the result, we can conclude that variable V_1 between 2 and 2.8 can be used to predict the number of collision slot efficiently. The result are in line with Schoute method where it indicated that 2.39 is an optimal value for V_1 .

5.2.3 Experiment Three Result

Figure 8 shows an average of ten runs of: a) number of slots, and b) number of frames, compared between VCEP method (variable $2.0 \leq V_1 \leq 3.0$, variable $0 < V_2 < 1$) versus Sch and LB methods. VCEP uses both variable V_1 and V_2 to predict collision slot and empty slot. VCEP method also includes variables used in VEP method, where $V_1 = 2$ and $0 < V_2 < 1$. Due to the space availability, figure 8 only shows those results for variables that have better outcome compared to Sch and LB methods. From the figure, we can see there are thirty variables out of ninety-nine variables that used either less number of slots or number of frames compared to Sch and LB methods.

Figure 7: Comparison between VCP method with different variables versus Sch and LB methods: a) Number of slots, b) Number of frames.

There are four major variables with visible improvement in total slots and frame counters, these variables are: 1) $V_1 = 2, V_2 = 0.1$, 2) $V_1 = 2, V_2 = 0.2$, 3) $V_1 = 2.1, V_2 = 0.1$, and 4) $V_1 = 2.2, V_2 = 0.1$. From the result we conclude that, any variable V_1 between 2 and 2.2 and variable V_2 between 0.1 and 0.2, can be used as an efficient Backlog estimation.

5.3 Analysis and Discussion

This sub-section compares and analyses all results from the three experiments. Figure 9 shows that different variables have impacted on total number of slots and frames. When only one variable V_1 is used to predict the number of collision slot (VCP), any variable between 2 and 2.8 is suitable (Shown as 'White' area in figure 9). On the other hand, when variable V_2 is taken into consideration, only few variables give the best results, these variables are 1) $V_1 = 2, V_2 = 0.1$, 2) $V_1 = 2, V_2 = 0.2$, 3) $V_1 = 2.1, V_2 = 0.1$, and 4) $V_1 = 2.2, V_2 = 0.1$. Any other lower

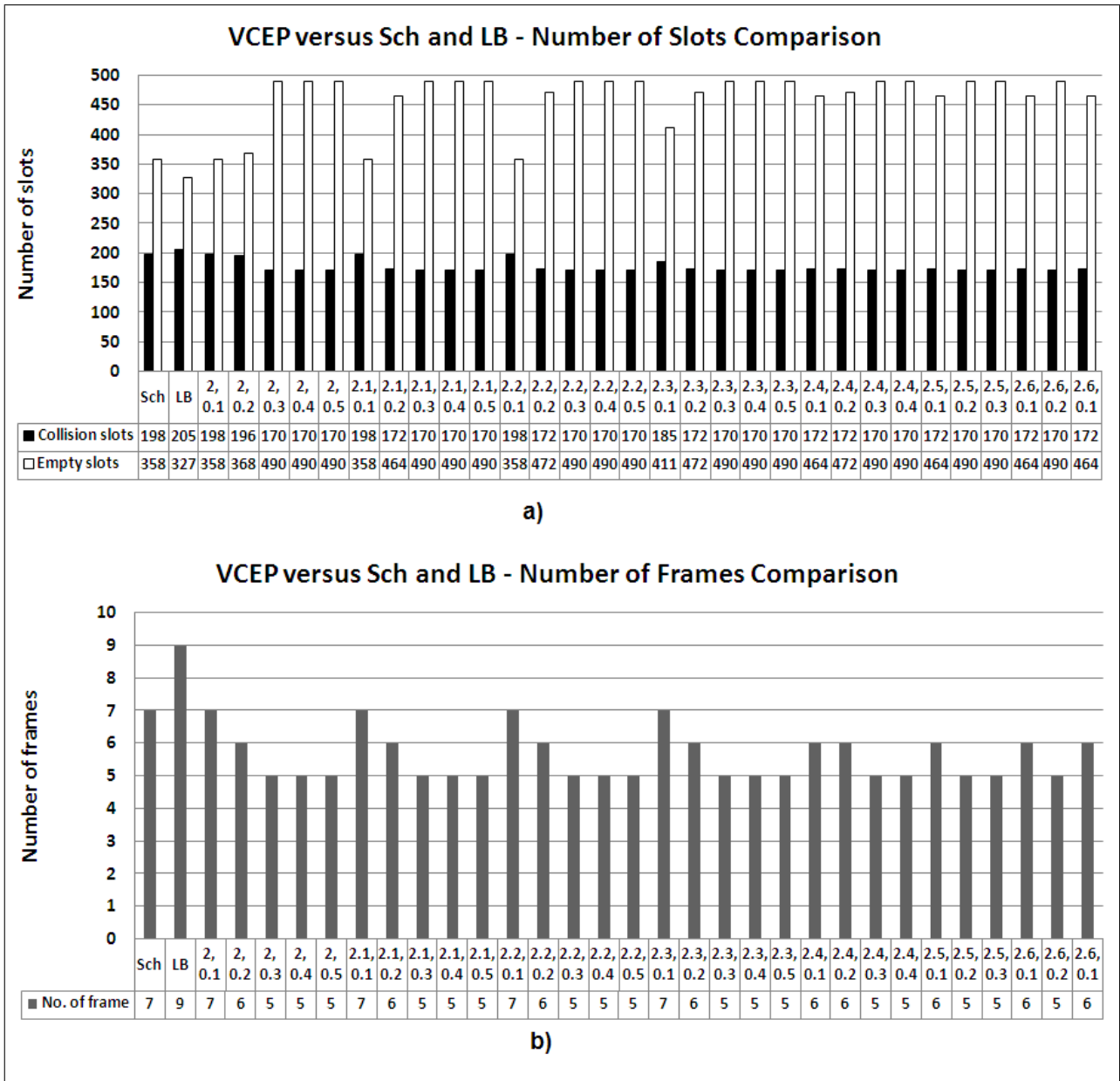


Figure 8: Comparison between VCEP method with different variables versus Sch and LB methods: a) Number of slots, b) Number of frames.

variables give various results; some with less number of slots and some with less number of frames compared to Sch and LB methods (Shown as ‘Grey’ area in figure 9). The ‘Black’ area in figure 9 represents those variables with poor results where total number of slots and frames are higher than Sch and LB.

The analysis clarifies that there are more than one suitable variables to predict Backlog; and that these parameters involve both collision slots and empty slots. From the analysis and discussion, we conclude that the DFSA performed more efficiently compared to Sch and LB when the parameter V_1 and V_2 used are low numbers. In order to minimise number of frames and slots, the variable V_1 should be between 2 to 2.8 for VCP method; the variable V_2 should be between 0.1 to 0.5 for VEP method; and the variable V_1 should be between 2 to 2.7; and the variable V_2 should be between 0.1 to 0.5 for VCEP method. However, total number of slots and frames also depends on the combination of the two parameters for VCEP (including VEP) method. Both parameters should be one low and one high or both low, to optimise the scheme.

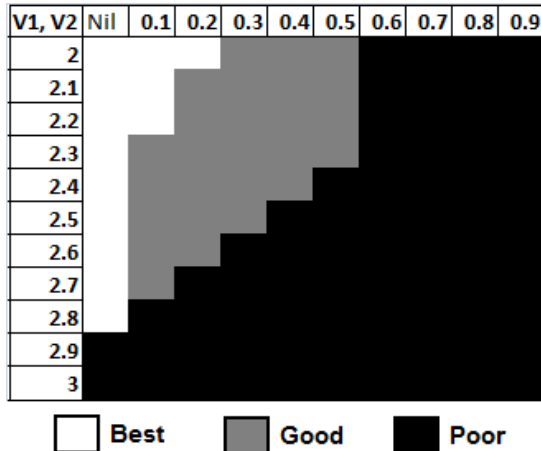


Figure 9: Result matrix of VEP, VCP, and VCEP methods using different variables compared to Schoute and Lowerbound methods.

6 Conclusion

In this work, we investigated the significance of RFID tags anti-collisions and developed efficient tag estimation method to improve the system efficiency of DFSA. We proposed a “Precise Tag Estimation Scheme”, which estimate precise number of tags around the reader.

The results and analysis have indicated that various parameter used by “Precise Tag Estimation Scheme”, including *empty slots* variables and/or *collision slots* variables, have an impact on system efficiency. Hence, variables used in VEP, VCP, and VCEP methods can change number of frames and slots, and the smaller values of both parameters V_1 and V_2 have a better impact on system efficiency.

In terms of future work, we intend to further test “Precise Tag Estimation Scheme” on different data sets with different number of tags. Different initial Q parameter will also be investigated to determine the impact on number of total frames and slots.

Acknowledgements

This research is partly supported by ARC (Australian Research Council) grant no DP0557303.

References

- Abramson, N. (1970), The ALOHA System - Another Alternative for Computer Communications, in ‘Proceedings of Fall Joint Computer Conference, AFIPS Conference’, Houston, Texas, pp. 281–285.
- Brown, M., Patadia, S. & Dua, S. (2007), *Mike Meyers’ Certification Passport: CompTIA RFID+ Certification*, McGraw-Hill.
- Cha, J. R. & Kim, J. H. (2005), Novel Anti-collision Algorithms for Fast Object Identification in RFID System, in ‘ICPADS ’05: Proceedings of the 11th International Conference on Parallel and Distributed Systems - Workshops’, IEEE Computer Society, Washington, DC, USA, pp. 63–67.
- Chen, W. T. (2006), ‘An Efficient Anti-Collision Method for Tag Identification in a RFID System’, *IEICE Transactions* **89-B**(12), 3386–3392.
- Cho, H., Lee, W. & Baek, Y. (2007), LDFSA: A Learning-Based Dynamic Framed Slotted ALOHA for Collision Arbitration in Active RFID Systems, in ‘Advances in Grid and Pervasive Computing Second International Conference’, Vol. 4459, Springer Berlin/Heidelberg, Paris, France, pp. 655–665.
- Choi, J. H., Lee, H. J., Lee, D., Lee, H. S., Youn, Y. & Kim, J. (2008), ‘Query Tree Based Tag Identification Method in RFID Systems’. www.freshpatents.com/Query-tree-based-tag-identification-method-in-rfid-systems-dt20080508ptan20080106383.php.
- EPCGlobal (2006), ‘EPCGlobal Tag Data Standards Version 1.3: Ratified Specification’. <http://www.epcglobalinc.org/standards/tds/>.
- Floerkemeier, C. (2007), Bayesian Transmission Strategy for Framed ALOHA Based RFID Protocols, in ‘RFID, 2007. IEEE International Conference on RFID Gaylord Texan Resort’, Grapevine, TX, USA, pp. 228–235.
- Jain, S. & Das, S. R. (2006), Collision avoidance in a dense RFID network, in ‘WINTECH ’06: Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization’, ACM, New York, NY, USA, pp. 49–56.
- Landt, J. (2001), *Shrouds of Time The history of RFID*, An Aim Publication, Pittsburg, PA.
- Lee, C. W., Cho, H. & Kim, S. W. (2008), ‘An Adaptive RFID Anti-Collision Algorithm Based on Dynamic Framed ALOHA’, *IEICE Transactions* **91-B**(2), 641–645.
- Lee, S. R., Joo, S. D. & Lee, C. W. (2005), An enhanced dynamic framed slotted aloha algorithm for rfid tag identification, in ‘MOBIQUITOUS ’05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services’, IEEE Computer Society, Washington, DC, USA, pp. 166–174.
- Li, B., Yang, Y. & Wang, J. (2009), ‘Anti-collision Issue Analysis in Gen2 Protocol - Anti-collision issue analysis considering capture effect’, *Auto-ID Labs White Paper*. <http://www.autoidlabs.org/single-view/dir/article/6/320/page.html>.
- Myung, J. & Lee, W. (2006a), ‘Adaptive binary splitting: a RFID tag collision arbitration protocol for tag identification’, *Mob. Netw. Appl.* **11**(5), 711–722.
- Myung, J. & Lee, W. (2006b), Adaptive splitting protocols for RFID tag collision arbitration, in ‘MobiHoc ’06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing’, ACM, New York, NY, USA, pp. 202–213.
- Quan, C. H., Hong, W. K. & Kim, H. C. (2006), Performance Analysis of Tag Anti-collision Algorithms for RFID Systems, in ‘Emerging Directions in Embedded and Ubiquitous Computing’, Vol. 4097, Springer Berlin/Heidelberg, Seoul, Korea, pp. 382–391.
- Ryu, J., Lee, H., Seok, Y., Kwon, T. & Choi, Y. (2007), A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID systems, in ‘MobiHoc ’06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing’, IEEE Computer Society, Glasgow, UK, pp. 5981–5986.
- Schoute, F. C. (1983), ‘Dynamic Frame Length ALOHA’, *IEEE Transactions on Communications* **31**(4), 565–568.
- Shin, J. D., Yeo, S. S., Kim, T. H. & Kim, S. K. (2007), Hybrid Tag Anti-collision Algorithms in RFID Systems, in ‘Computational Science ICCS 2007’, Vol. 4490, Springer Berlin/Heidelberg, Beijing, China, pp. 693–700.
- Vogt, H. (2002), Efficient Object Identification with Passive RFID Tags, in ‘Pervasive ’02: Proceedings of the First International Conference on Pervasive Computing’, Springer-Verlag, London, UK, pp. 98–113.
- Wang, Z., Liu, D., Zhou, X., Tan, X., Wang, J. & Min, H. (2007), ‘Anti-collision Scheme Analysis of RFID System’, *Auto-ID Labs White Paper*. <http://www.autoidlabs.org/single-view/dir/article/6/281/page.html>.