Improving Reader Performance of an UHF RFID System Using Frequency Hopping Techniques

Ju-Yen Hung and Venkatesh Sarangan*, MSCS 219,
Computer Science Department, Oklahoma State University, Stillwater, OK 74078

Abstract— **In this paper, we propose a new RFID passive tag reading model using frequency hopping techniques to reduce external interference as well as the number of collisions during the reading process, so that the overall tag reading performance is improved**.

I. Introduction

In UHF RFID systems, a tag transmits its information using "backscatter" technology [1]. If some tags within the same interrogation zone backscatter at the same time, the modulated waveforms will be garbled and no information will reach the reader. This is so called collision problem [2]. The limited computation ability of a tag made it hard to communicate among tags to avoid collisions. Instead, the reader takes the responsibility. Nevertheless, the collision problem is hard to avoid and has greatly influenced the reader's performance.

In spread spectrum communications, when the carrier frequency of a transmitted signal is periodically changing in time intervals, it is called frequency hopping [3]. (Figure 1)

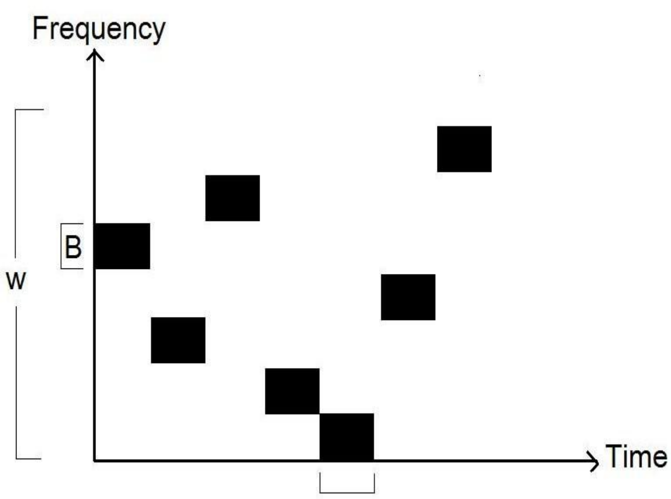

Fig. 1. Frequency-hopping pattern [3]. The duration of a hop interval is called hop period. Each frequency in a hopset (the set of hopping frequencies) has a bandwidth B. A hopset's bandwidth called hopping band, denoted W, is greater or equal to the number of frequencies in the hopset times B.

*The authors can be reached at juyen@cs.okstate.edu and saranga@cs.okstate.edu.

A frequency hopping system has the potential of avoiding interference [4]. If one of the frequencies was jammed, a frequency hopping system lost fewer information bits due to the signals being several times widely spread over its original spectral [5]. However, when hopping from one frequency to another frequency, there is a small amount of "switching time" in which no information will be transmitted [3]. Frequently changing of frequencies produces more such switching overhead, even though the impact of interference is reduced as well. Wisely choosing frequency hopping rate can reduce both the overhead and retransmission time.

II. Previous work

RFID anti-collision protocols can be generally classified as deterministic algorithms and probabilistic algorithms.

Deterministic algorithms, also known as tree based algorithms, prevent collisions by muting most of the involved tags. Eventually, there will be a successful transmission from a tag [6]. The reader finished reading all tags in its interrogation zone by visiting them one by one. The advantage of tree algorithms is that the system can obtain higher accuracy, but takes a longer time to read all tags when compared to probabilistic algorithms, especially when a huge number of tags are present at the same time. On the other hand, probabilistic algorithms, including the family of ALOHA based protocols, can read a larger number of tags in a shorter time but in a less accurate manner. There are a lot of extended slotted ALOHA algorithms, some of the most popular will be discussed in the following sections.

A. Framed Slotted ALOHA protocols

Framed-Slotted ALOHA (FSA) is the most well-known protocol among all deterministic algorithms [7]. By letting each tag transmitting its information to a randomly chosen time slot in a frame, FSA reduces the probability of tag collision. However, if the difference between the frame size and the number of tag counts is large, either idle slots or the number of collisions are also large. This highly degrades the system's efficiency.

Dynamic FSA (DFSA) [8] and Adaptive Slotted ALOHA Protocol (ASAP) [14] solve this problem by estimating the number of tags present to determine the ideal frame size in the subsequent round. In DFSA, if the tag counts are large, because no matter how many tags remaining unread, it always starts with the initial minimum frame size after identifying a tag [9], the frame size needs to be exponentially increased. In ASAP, the frame size is determined based on the observation of the previous round. These algorithms work well if the tag counts are small. However, the performance is poor if the quantity becomes large [9] [10], because the frame size cannot increase indefinitely as the tag counts increase and the fact that large frame sizes increase the interference between readers in multiple-reader environments. As a result, we need a scheme that can minimize the reading time even if the frame size is limited.

Enhanced DFSA (EDFSA) [11] guarantees a high tag reading rate with a limited frame size by grouping tags to a smaller population, so that the probability of a successful reserved slot can be maintained close to 36.8% of the maximum frame size [12]. This approach, however, does not significantly reduce the rounds needed for reading tags.

B. Accelerated Framed Slotted ALOHA (AFSA)

The framework of AFSA [12] extends the three phases seen in most slotted ALOHA protocols to five phases (Figure 2). The first phase is the *advertisement phase*, where the reader broadcasts to all tags within its range: the frame size (N), the number of groups (M) and an *n*, which represents the length of an n-bit sequence used for the next phase. A tag first randomly chooses its group number to determine its eligibility to participate in the proceeding round. Each eligible tag then changes its state to "*select*", and chooses randomly a time slot.

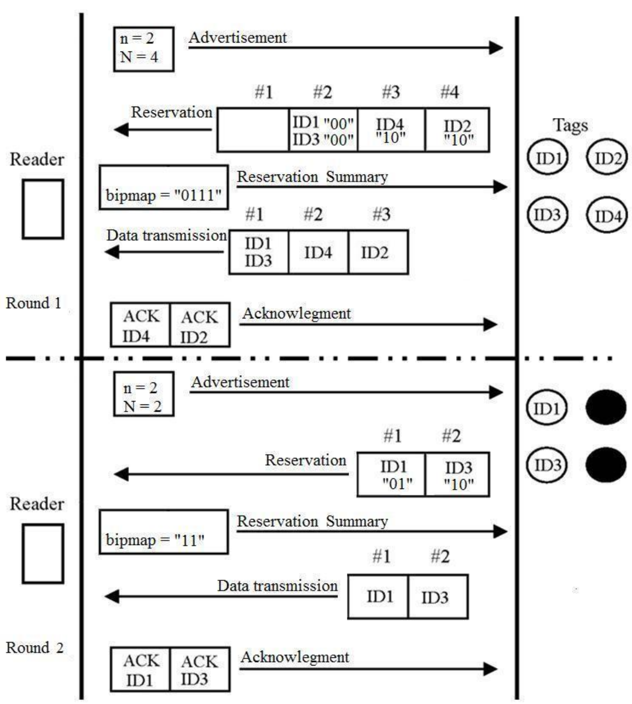

Fig. 2. A five-phase AFSA model, where M = 1.

The second phase is the *reservation phase*, during which each tag transmits an n-bit sequence in its chosen slot. There are 2^n possible n-bit sequences, according to the value of *n* advertised in the previous phase. If the reader receives an n-bit sequence during a time slot, it assumes there is some tag that has successfully reserved that time slot for transmitting its data. If a garbled signal is received, the reader knows there is a collision between two or more tags in that slot.

The third phase is the *reservation summary phase*, in which a bitmap is generated to inform the slot reservation status for tags. A 0 in the i^{th} position of this N-bit summary bitmap indicates either no tag has reserved the i^{th} time slot or a collision occurred in that slot. Nevertheless, a 1 does not guarantee only one tag has chosen that slot. If more than one tag has chosen the same time slot and has transmitted the same n-bit sequence to make the reservation, the reader cannot detect the collision and when those tags transmit data in the later phase, those tags cause a collision. This is called *undetected collision*.

The fourth phase is the *data transmission phase*, wherein all tags that find themselves as successfully reserved statuses transmit their data in the order of the counting of 1s until its position on the bitmap. For example, if the summary bitmap is 0110, the tag that reserved the third time slot should transmit its data second. The rest will go back to "active" state and wait for the next advertisement.

The last phase is the *acknowledgment phase*. The reader acknowledges the data transmission from the tags in the form of bits; 0 denotes a failure, 1 denotes a success. A tag receiving a positive acknowledgment will mute itself. Otherwise it goes back to "active" state and waits for the next advertisement.

The above five phases are executed sequentially. In order to minimize the average round time, the value of *n* is limited in the size so that the time for reservation will not be prolonged.

C. Advantages and drawbacks for AFSA

AFSA reduces the number of idle slots as well as the number of collisions so that the average tag reading time is reduced by up to 40% with respect to the stand alone ALOHA protocols [12]. It is also found from the results of the simulation that the optimal value of *n* is 2, which minimizes the total round time when the N and K are known, where K is the participated tag counts for each round. However, by using *n* = 2, we can at most have four different n-bit sequences which produce a large number of undetected collisions that lead to a waste of time slots in the data transmission phase. If we can increase the value of *n* without increasing the total round time, the undetected collision can be reduced and thus improves the performance of the reader.

In our new model, we adapted all assumptions as to AFSA. We are aiming on reducing the retransmission time caused by external interference and the average tag reading time by minimizing the number of undetected collisions.

III. Reduce transmission time

If a piece of information is transmitted over the air as a whole, it is more efficient. But if some interference is taking place during the transmission, information will be garbled. Those lost bits need to be retransmitted to recover the information. A slow frequency hopping system provides interference resistance by nature. During each hopping period a portion of the information will be transmitted. If some channel is jammed or intercepted, the lost information is limited to the portion using that frequency, not the whole piece of information [5] [13]. The faster hopping rate seems to have better interference resistance, but produces more switching overhead, which possibly makes the system less efficient.

Assume some b-bit information, which is divided into m portions and modulated to m chips during transmission, each chip period is T_c , where

$$T_c = \delta + b/mR \quad (1)$$

δ is the switching overhead, R denotes data rate, b/mR is the time that transmits signals (dwell time). Assume interference occurs at the beginning of transmitting i^{th} chip and continue for T_i seconds. The time for retransmitting the lost bits is kT_c, where

$$k = \text{ceiling}\,(T_i / T_c) \quad (2)$$

Let p denote the percentage of total jammed channels, where $0 \le p \le 1$. The total time spent for reading one tag with retransmitting lost bits is justified as

$$T = mT_c + kT_c * p = T_c\,(m+kp) = (\delta + b/mR)\,(m+k) \quad (3)$$

$$\text{Throughput } S = b / [(\delta + b/mR)\,(m+kp)] \quad (4)$$

Depending on the probability of occurring interferences, we found that by using optimal value of $m^* = \sqrt{\frac{128kp}{10}}$, the maximal throughput can be achieved.

IV. Reduce Undected Collissions

In the previous study, AFSA executes the 5 phases sequentially. With frequency hopping techniques, we are able to execute these 5 phases in a two-stage pipeline scheme. To implement this model, the reader must be able to monitor both uplink and downlink channels. In other words, the reader should be full duplex, which provides the functionality to transmit and receive data simultaneously. Figure 4 shows an AFSA model without frequency hopping. Figure 5 shows an AFSA model with frequency hopping.

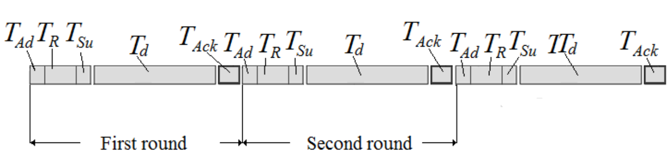

Fig. 4. Accelerated Framed Slotted ALOHA, sequential execution of five phases.

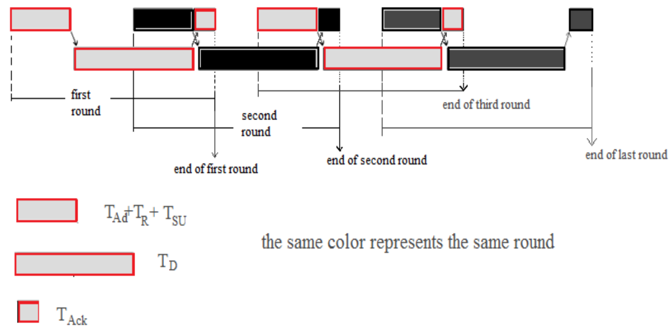

Fig. 5. Execute the five phases of AFSA in a pipelined scheme

Let T_{AD}, T_R, T_{SU}, T_D, and T_{ACK} denote the duration for advertisement, reservation, summary, data transmission, and acknowledgment phases respectively (all in μs). We have

$$\begin{aligned} T_{AD} &= 12.5 * (20 + \log_2 M + n) \\ T_R &= 12.5 * N\,(n+1) \\ T_{SU} &= 12.5 * (10+N) \\ T_D &= S\,(80 * 4 + 12.5) \\ T_{ACK} &= 12.5 * (10 + S) \end{aligned} \quad (5)$$

where S is the number of available time slots for data transmission phase.

From [12], we know $S \cong 0.38N$ and $n = 2$ have the best efficiency when executing sequentially. Let T_{SEQ} denote the total time of a round for sequential scheme and T_{HOP} for pipelined scheme. T_{SEQ} can be written as

$$T_{SEQ} = T_{AD} + T_R + T_{SU} + T_D + T_{ACK} \quad (6)$$

Since pipelining will take effect when there is more than one round, we assume the reading takes i rounds. On average, T_{HOP} is

$$T_{HOP} = (T_{AD} + T_R + T_{SU} + T_D + T_{ACK} + (i-1) * T_D)/i$$

$$= T_D + (T_{AD} + T_R + T_{SU} + T_{ACK})/i < T_{SEQ} \quad (7)$$

We know that n announced in advertisement phase is the key factor of occurring undetected collision in the reservation phase. As n increases, the probability of undetected collisions reduces, but durations of advertisement and reservation phases increase. We also noticed that as long as this increasing amount of time is small enough, that is, if

$$T_{AD} + T_R + T_{SU} + T_{ACK} \approx T_D \quad (8)$$

we can maximize the throughput. From above, the Optimized $n^* = \frac{7.728N - 4}{N+1}$ can both reduce the number of undetected collisions as well as total read rounds, and further improve the reader performance.

V. Simulation Results

The simulations are done in Java and the results presented in this section are the outcomes of 50 different runs. The testing is divided into two portions, first part tests our new model with different *n* values, where *n* = 5~8. Each *n* value tests for 50 times with increment of 500 tags and is executed until the unread tag counts less than 2 to provide 99% accuracy. The second part tests and compares AFSA between pipeline scheme and sequential scheme. Each scheme tests for 50 times with increments of 50 tags and is executed until the unread tag counts less than 2 to provide 99% accuracy. In this part, interference is also considered to be possible and the probability of interference is generated randomly by the program. For simplicity, a tag will retransmit all its information in case of interference.

As a result of simulations, we have found that using *n* = 6 in the pipelined scheme protocol minimized the total round time for given number of time slots (Table I and Figure 5).

TABLE I
Total Time Spent with Different n Values

	Total time spent (second)			
tag count	n=5	n=6	n=7	n=8
500	0.189558	0.19144	0.19831	0.2035
1000	0.387063	0.366668	0.366625	0.3785
1500	0.567943	0.53857	0.536013	0.555627
2000	0.76234	0.684205	0.711415	0.76786
2500	0.89745	0.861428	0.88501	0.926433
3000	1.037185	1.033648	1.047045	1.109575
3500	1.23793	1.231615	1.225283	1.330568
4000	1.411162	1.405355	1.43062	1.445057
4500	1.569765	1.52855	1.5463	1.653975
5000	1.73402	1.725595	1.72965	1.852017
:	:	:	:	:
:	:	:	:	:
20500	7.109457	6.90744	7.056878	7.581985
21000	7.140802	7.194222	7.2058	7.565645
21500	7.427303	7.308308	7.35098	7.822688
22000	7.74562	7.453395	7.51891	8.019738
22500	7.97172	7.763678	7.656093	8.164128
23000	8.048528	7.905148	7.882303	8.377013
23500	8.11373	7.983488	8.052575	8.59308
24000	8.345117	8.124653	8.288745	8.874668
24500	8.614912	8.27683	8.39088	9.007875
25000	8.603815	8.463832	8.582523	9.084235

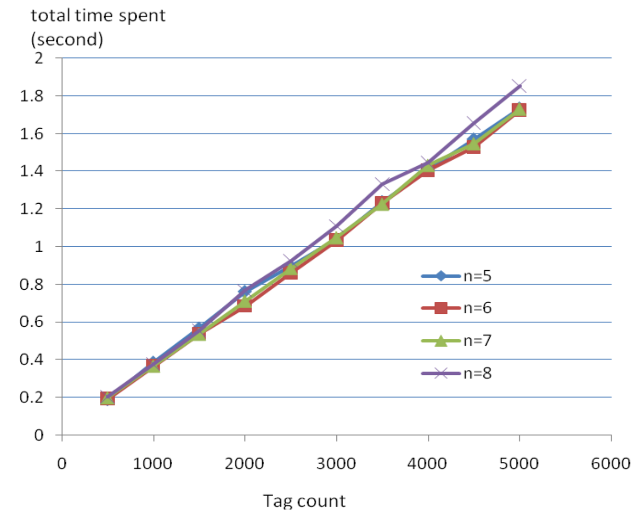

Fig. 6. Total time spent with different *n* values.

The tests of pipelined scheme and sequential scheme are using different *n* values. For pipelined scheme, *n* = 6, which is based on the results of the first part testing; the sequential scheme uses *n* = 2, for it has been proved to be the optimal value for AFSA. We list the results of reading 50~2500 tags using both schemes in Table II.

TABLE II
Average Reading Time Using Pipelined and Sequential Scheme

	Average tag reading time	
Tag counts	Pipelined scheme	Sequential scheme
250	741.18	1769.06
500	725.715	1648.35
750	632.98	1550.753
1000	622.2925	1759.085
1250	724.406	1923.408
1500	776.64	1947.363
1750	669.1257	2119.763
2000	716.0625	2106.715
2250	707.7522	1549.613
2500	639.417	2276.322
2750	683.2455	1633.504
3000	727.115	1550.742
3250	658.8762	1960.914
3500	662.4093	1744.2
3750	734.3507	1710.655

It is obvious that on average the pipelined scheme is twice as fast as sequential scheme. Figure 7 shows two very different lines. The pipelined results produce a smoother line, which means it is less influenced by interference; on the other hand, the sequential scheme suffered greatly through interference so that the produced line jumped violently. It proved that the pipelined scheme was more interference resistant and more efficient than the sequential scheme.

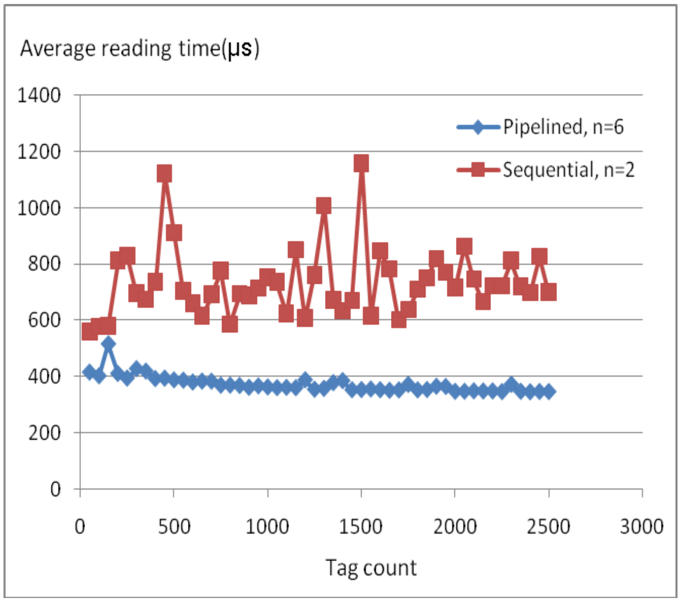

Fig. 7. Average reading time using pipelined and sequential scheme

Figure 8 shows the comparison between pipelined and sequential scheme over average tag reading time.

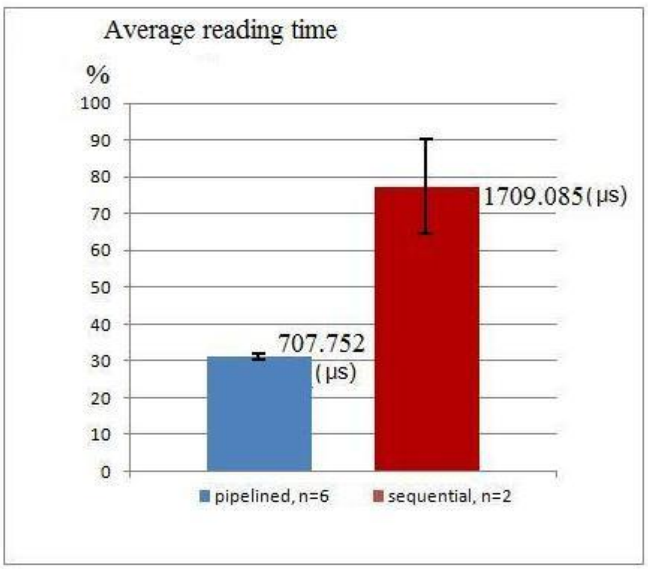

Fig. 8. Comparison between pipelined and sequential scheme over average tag reading time

VI. Conclusion

The impressive performance of our new model, not only high interference resistance but also high collision avoidance, has proven to increase efficiency by 50 percent on average, compared with sequential execution of AFSA. The key factor is that we execute simultaneously the four phases that are less time consuming with the data transmission phase, which is taking twice as much execution time as the sum of the other four phases. Furthermore, we filled up the time gap between the two pipelined stages with a longer n-bit sequence, which eliminated most undetected collisions.

We have proved that with frequency hopping techniques the influence of external interference can be minimized. We also use a two-stage pipeline scheme to cut down the total communication time between reader and tags. In the future, the same scheme can be deployed in mobile environments, though it will be a more complex and challenging work.

References

[1] K. Finkenzeller, *RFID Handbook*, 2nd ed., New Jersey: Wiley & Sons, 2003, pp. 29–56, 183-192, 311-314.

[2] P. Sorrells, (2008, November 11) *Passive RHID basics.* Microchip Technology Inc., 1998 [online]. Available: http://ww1.microchip.com/downloads/en/AppNotes/00680b.pdf

[3] D. Torrieri, *Principles of Spread Spectrum Communication Systems*, New York: Springer, 2005, pp. 193-384.

[4] W. Stallings, *Wireless Communications and Networks*, New Jersey : Prentice Hall, 2001, pp. 131-197, 479-520

[5] R.M. Buehrer, (2008, December 21) *Spread Spectrum Communications*, 2007[online]. Available: http://www.mprg.org/people/buehrer/5660/Lectures.htm.

[6] W. Chen and G. Lin. "An efficient Anti-Collision Method for Tag Identification in a RFID System" in *IEICE Transactions on Communications*, Vol. E89-B, No. 12 December 2006, pp. 3386-3392.

[7] *EPCTM Radio-Frequency Identification Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz*, Version 1.0.9, EPCglobal, Jan. 2005.

[8] J. Cha, J. Kim, "Dynamic Framed Slotted ALOHA Algorithms using Fast Tag Estimation Method for RFID System", in *Proc. IEEE CCNC 2006*, pp. 768-772.

[9] S. Lee, S. Joo, and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," in *Proc. MobiQuitous*, pp. 166-172, July 2005.

[10] H. Vogt, "Multiple Object Identification with Passive RFID Tags." In *2002 IEEE International Conference on Systems, Man and Cybernetics.* October 2002.

[11] H. Vogt, "Efficient Object Identification with Passive RFID Tags." In *International Conference on Pervasive Computing*, LNCS. Springer-Verlag, 2002. pp. 98-113.

[12] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A framework for fast RFID tag reading in static and mobile environment." *Computer Networks journal*, *52*, 5 (April 2008), pp. 1058-1073.

[13] M. K. Simon, J. K. Omura, R. A. Scholtz and B. K. Levitt, *Spread Spectrum Communications Handbook*. New York: McGraw-Hill, Inc., 2002, pp. 20-29.

[14] G. Khandelwal, K. Lee, A. Yener, and S. Serbetli, "ASAP: a MAC Protocol for Dense and Time-Constrained RFID Systems," *EURASIP J. Wireless Commun. and Networking*, vol. 2007, article ID 18730, 13 pages, 2007. doi:10.1155/2007/18730.