

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 18730, 13 pages
doi:10.1155/2007/18730

Research Article

ASAP: A MAC Protocol for Dense and Time-Constrained RFID Systems

Girish Khandelwal,¹ Kyoungwan Lee,² Aylin Yener,² and Semih Serbetli³

¹Qualcomm, San Diego, CA 92121, USA

²Wireless Communications and Networking Laboratory, Department of Electrical Engineering, Pennsylvania State University, University Park, PA 16802, USA

³Philips Research, 5621 Eindhoven, The Netherlands

Received 16 October 2006; Revised 10 March 2007; Accepted 21 June 2007

Recommended by Alagan Anpalagan

We introduce a novel medium access control (MAC) protocol for radio frequency identification (RFID) systems which exploits the statistical information collected at the reader. The protocol, termed adaptive slotted ALOHA protocol (ASAP), is motivated by the need to significantly improve the total read time performance of the currently suggested MAC protocols for RFID systems. In order to accomplish this task, ASAP estimates the dynamic tag population and adapts the frame size in the subsequent round via a simple policy that maximizes an appropriately defined efficiency function. We demonstrate that ASAP provides significant improvement in total read time performance over the current RFID MAC protocols. We next extend the design to accomplish reliable performance of ASAP in realistic scenarios such as the existence of constraints on frame size, and mobile RFID systems where tags move at constant velocity in the reader's field. We also consider the case where tags may fail to respond because of a physical breakdown or a temporary malfunction, and show the robustness in those scenarios as well.

Copyright © 2007 Girish Khandelwal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Radio frequency identification (RFID) systems provide an efficient and inexpensive mechanism for automatically collecting the identity information of an object [1, 2]. In a system, tags with unique identities communicate with an RFID reader over a wireless multiaccess channel [3, 4]. Recently, there has been an intense effort towards the development of RFID systems for their many promising applications from providing security to factory automation to supermarket checkouts [5, 6]. All of these envisioned applications call for a need to deploy a large number of tags in small geographical areas and have the tags autonomously communicate with the reader(s). As such, RFID systems of the near future will be dense wireless networks with limited radio resources that will have to be shared by the tags via contention-based methods. Further, these systems will be considered operational when most or all of the tags in a reader field are successfully identified in a short amount of time.

In such a network setting, the design of an efficient MAC protocol is of paramount importance. The performance degrading impact of excessive collisions in random multi-

access communications is well known [7, 8]. Indeed, tag collisions, which occur when multiple tags simultaneously transmit information in the same channel, severely limit the performance of RFID systems. In this paper, we will focus on alleviating this limitation via intelligent MAC design.

In recent years, many attempts have been made to confront the tag-collision problem. The methods suggested for RFID systems up to date can be classified into two categories: variants of ALOHA that rely on randomizing the access times of tags to reduce collisions; and tree search methods that aim to avoid collisions and identify one tag at a time. STAC, based on slotted ALOHA, has been proposed in [3, 9] for Class-1 Generation-1 RFID systems and binary tree search has been proposed for Class-0 Generation-1 RFID systems [4, 10].

In the binary tree search algorithm, one tag is identified at a time without a collision. In contrast, STAC is more likely to lead to severe tag collisions if the frame size is not properly chosen. In order to avoid this severe performance loss, frame size adaptive MAC protocols for RFID system were proposed in [11–15]. The frame size adaptive MAC protocol in [11] uses a simple estimate for the tag population in each round (frame) in order to adaptively adjust the frame size

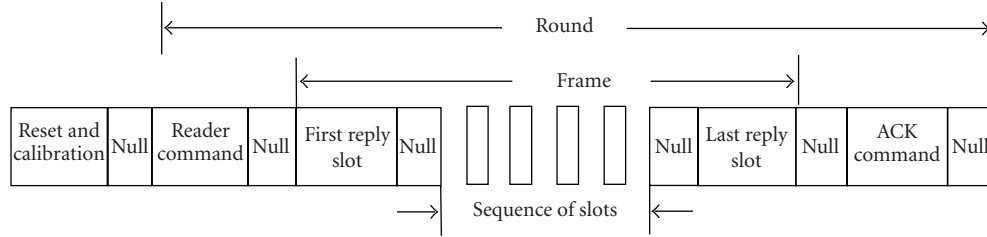


FIGURE 1: Round structure of ASAP.

in the subsequent round based on the minimization of the time required to identify all tags with a given level of assurance. In [12, 13], the frame size was found to maximize the expected throughput of framed ALOHA. To find the frame size, the probability distribution of the number of tags transmitting is obtained by adopting the Bayesian approach outlined in [16]. The another frame size adaptive MAC protocol for both passive and active RFID tags was developed in [14]. The more recently proposed Class-1 Generation-2 also provides the option of a variable frame size [15]. Even though these attempts provide a notable performance improvement over fixed frame size RFID MAC protocols, they may still lead to less than acceptable performance for dense RFID systems.

We note that the foregoing research work focuses on resolving the tag collision problems in RFID systems where multiple tags communicate with a single reader over a shared wireless medium. When the multiple readers communicate with multiple tags, the reader collision might occur if an RFID reader interferes with the operation of another reader. There is considerable research effort towards developing anticollision algorithms for the reader collision problem [17, 18]. In [17], a simple and distributed time division multiple access (TDMA) reservation anticollision algorithm was developed. The attempt to find the optimum solution for the reader collision problem was made based on a hierarchical Q-Learning algorithm [18].

In this paper, we propose a novel MAC protocol for RFID systems that have a large number of passive tags. The underlying motivation is to design a MAC protocol, that is, compatible with the suggested standards in [3, 15], and to obtain substantial improvement in read-time performance as compared to existing methods, for example, [11–15]. As is the case for EPC global standards [3, 15], the proposed adaptive slotted ALOHA protocol (ASAP) is based on framed slotted-ALOHA [19, 20] and aims to reduce the probability of tag collisions while simultaneously expediting the identification of RFID tags. The key is to efficiently utilize the statistical information inherently collected at the reader to determine the next frame size.

The design of ASAP entails an ML-based estimation algorithm for the number of tags to be identified with the frame size decision algorithm designed to optimize an efficiency function defined in the sequel. We also extend the design of ASAP to handle more realistic RFID systems. To that end, we first consider the case where the frame size is limited (p -ASAP). Next, mobile RFID systems (m -ASAP) where tags

move in the reader’s field are considered. In particular, for the mobile scenario, we aim to determine the maximum tag arrival rate while providing a statistical guarantee for the percentage of the tags read during their presence in the reader field. Finally, we consider the case where the tags may not respond due to a physical breakdown or a temporary malfunction. We demonstrate that ASAP has impressive performance in all scenarios we consider for dense RFID systems, and outperforms previously proposed MAC protocols including [11].

2. SYSTEM MODEL AND MECHANICS OF ASAP

We consider the collision limited¹ 900 MHz UHF RFID system where large number of passive tags try to communicate with one reader over a shared channel. We assume that each passive tag transmits a data packet with a symbol duration of $4 \mu\text{s}$ [3, 4].

Reader to tag communication is accomplished using “0,” “1,” and “Null” data symbols as defined in [4]. The reader uses “0” and “1” to form commands, and “Nulls” to signify the beginning of a command, the end of a command, and to close the slots within a frame. The reader transmits data in the former portion of the $12.5 \mu\text{s}$ symbol duration [4]. Communications between the reader and the tags take place in rounds whose structure is shown in Figure 1. This structure is compatible with STAC [3] as well as the EPC global Class-1 Generation-2 [15].

To explain the communication between the reader and the tags, consider tag state machine described in Figure 2. Initially, the tags are in an inactive “unpowered” state and they transition to the “activated” state, when they “listen” to the “reset,” the “oscillator calibration signals,” and the “data symbol calibration signals” as defined in [4]. The “reader command” provides the frame size for the ongoing round. The tags in the “activated” state collect the frame size information and transition to the “select and transmit” state. In this state, each tag randomly selects a slot for transmission and transmits its packets.

“Null” signals the completion of a command and the end of every slot in a frame. This facilitates resynchronization of the tags with the slot boundaries and allows the tags to keep

¹ The received SNR is shown to be high enough to justify this assumption with passive tags communicating in a short range in [21].

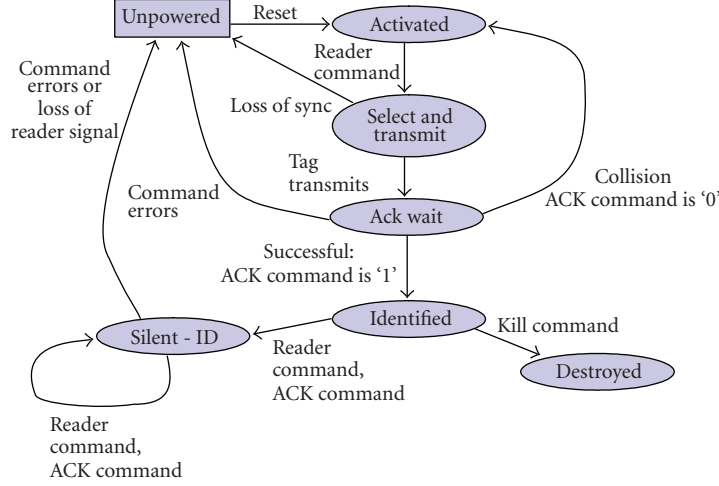


FIGURE 2: ASAP: tag state machine.

track of the slot number in the current frame. The duration for the detection of an idle slot is 10 data symbols.

Tags go to “ack wait” state after sending their identification strings. The reader transmits an “ACK command” at the end of the round. The length of the command varies in proportion to the frame size of the round. The reader transmits “1” if the transmission in the corresponding slot was successful. It transmits “0,” if the slot was either idle or the transmissions resulted in a collision. Positively acknowledged tags transition to the “identified” state and negatively acknowledged tags transition to the “activated” state. Subsequent to the transmission of the “ACK command,” the reader broadcasts a new “reader command” and a new round begins.

3. ASAP

ASAP proposes the optimum frame size for each round after estimating the number of tags present in the reader’s field. In each round, the reader begins with a “reader command” after the completion of the data calibration cycle shown in Figure 1. Primarily, it provides information about the frame size for the ongoing round. In this section, we discuss the design of the optimum frame size followed by a tag count estimation algorithm.

3.1. Design of the frame size

Consider first that the reader has already acquired the value of the tag count. We will explain how the reader obtains the ML estimate of the tag count later in the paper.

Define the duration of an idle slot T_I , and an occupied (successful or unsuccessful) slot T_B . Note that, we consider the case where $T_I \neq T_B$ in ASAP. In particular, we consider that the idle slots are closed prematurely, that is, $T_I < T_B$. Given these definitions, we consider an efficiency function, p_{eff} , defined as the ratio of expected time taken by the successful slots to the expected time taken by the idle and the unsuccessful slots, as our performance metric. The motivation behind defining such a metric is that maximizing this

function simultaneously increases the time due to successful transmissions, and decreases the time due to idle and unsuccessful transmissions, thus minimizing the waste of resources. We have

$$p_{\text{eff}} = \frac{E[S] \cdot T_B}{E[U] \cdot T_B + E[I] \cdot T_I}, \quad (1)$$

where $E[S]$, $E[I]$, and $E[U]$ are the expected number of *successful* slots, *idle* slots, and *unsuccessful* slots, respectively.

Given the (estimated) contending tag count in the reader’s field K , the problem is to *devise a frame size, N* , that maximizes the efficiency of the round p_{eff} . Since each tag independently selects any particular slot with equal probability, the expected number of successful, idle, and unsuccessful slots in a frame are given by

$$E[S] = K \left(1 - \frac{1}{N}\right)^{K-1}, \quad E[I] = N \left(1 - \frac{1}{N}\right)^K \quad (2)$$

$$E[U] = N - K \left(1 - \frac{1}{N}\right)^{K-1} - N \left(1 - \frac{1}{N}\right)^K. \quad (3)$$

Substituting (2)-(3), (1) becomes

$$p_{\text{eff}} = \frac{K}{N(1 - (1/N))^{1-K} - K + N(1 - (1/N))(\alpha - 1)}, \quad (4)$$

where $\alpha = T_I/T_B$.

Generally speaking, N can be chosen to an arbitrary function of K , that is, $N = f(K)$. In this paper, we consider a class of simple policies and assume that N is linearly related to K , that is, $N = \beta K$ and focus on finding the optimum multiplier. In this case, a closed form for the maximizer of p_{eff} can easily be found for large K . Using the approximation:

$$\lim_{K \rightarrow \infty} \left(1 - \frac{1}{\beta K}\right)^K \simeq e^{-1/\beta}, \quad (5)$$

simplifying (4), and discarding constant in the denominator, we obtain

$$P^{\text{eff}} = \frac{1}{\beta e^{1/\beta} + (\alpha - 1)\beta} \quad (6)$$

which is strictly pseudoconcave (see Appendix A for the proof). As a result, the local maximum is also the global maximum for $\beta > 0$ [22]. Then, the optimum frame size in each round is given by $N = \beta^* K$. We note that our design of the optimum frame size based on (6) can be readily used for any size of EPC and CRC memory bits. For the choice of 64-bit EPC and 16-bit CRC supported by Class-1 Generation-1 [3], $\alpha = T_I/T_B = 0.1884$ ($T_I = 40 \mu\text{s}$, $T_B = 320 \mu\text{s}$)². Therefore, we propose that the frame size in each round should be $N = \beta^* K = 1.943K$. For the 96-bit EPC and 16-bit CRC supported by Class-1 Generation-2, β^* is found to be 2.685.

The efficiency function can also be defined, by considering the total delay at the denominator, as follows:

$$P^{\text{eff}} = \frac{E[S] \cdot T_B}{E[U] \cdot T_B + E[I] \cdot T_I + E[S] \cdot T_B}. \quad (7)$$

We note that the same approximation (6) is obtained using (7) as well.

3.2. Tag count estimation algorithm

β^* was found under the assumption that the reader knows the tag count. In practice, the reader may not have the tag count, and has to estimate this parameter.

In ASAP, the tags respond with their identification strings in their chosen slots once in a round. Functionally, the reader collects tags' transmissions, performs cyclic redundancy checks, acknowledges successful identifications, and in the process, it inherently collects statistics on the total *idle* slot count (Z_I), the *successful* slot count (Z_S), and the *unsuccessful* slot count (Z_U). We propose to utilize this information to estimate the active tag count. In particular, we can use Z_I whose probability mass function (PMF) is given by [23]

$$P(Z_I = Y | N, K) = \sum_{i=0}^{N-Y} (-1)^i \binom{Y+i}{Y} \binom{N}{Y+i} \left(1 - \frac{Y+i}{N}\right)^K. \quad (8)$$

The ML estimation problem becomes

$$\hat{K}_{\text{ML}} = \underset{K \in \{K \geq Z_S + 2Z_U\}}{\text{argmax}} P(Z_I = Y | N, K). \quad (9)$$

The likelihood function in (9) can be enumerated for different K values to find its maximum. Note that we rely on Z_I and not Z_S for the ML estimation simply because the PMF of Z_S has local maxima [11].

TABLE 1: Tag count estimation in an identification process of ASAP.

Round number	1	2	3	4	5	6
Actual tag count	80	62	22	12	6	3
N	50	121	35	24	8	4
Z_I	10	72	19	16	4	1
Z_S	18	40	10	6	3	3
Z_U	22	9	6	2	1	0
Est. tag count for next round	62	18	12	4	2	0

In tag count estimation, one obvious concern is the range of K over which the likelihood function needs to be enumerated. We can use $K = Z_S + 2Z_U$ as the lower bound since we have ruled out the possibility of erroneous receptions in a slot occupied by a single tag as well as the capture effect. In this case, there are at least the number of successful tags plus twice the number of unsuccessful tags, because when there is an unsuccessful slot, at least two tags contend for the slot. We can also use the fact that for a given N and Z_I , the likelihood function has a unique maximum and it is a monotonically decreasing function for $K > \hat{K}_{\text{ML}}$. Thus, the search for \hat{K}_{ML} is stopped when the likelihood function value begins to decrease, for increasing K .

Even with this reduction in complexity, the two factorials in (8) may render the enumeration of the likelihood function computationally complex for large N and K . An alternative simpler estimator can be obtained by rearranging the expression in (2) for $E[I]$ and using $E[I] \approx Z_I$, as

$$\hat{K}_{\text{Exp}} = \frac{\log(Z_I/N)}{\log(1 - (1/N))}. \quad (10)$$

Table 1 shows the snapshot of a single identification process by employing our ML estimate algorithm and design of the frame size. The reader does not have any prior information of the tag count and it arbitrarily offers a frame size of 50 slots in the first round. We observe that the estimated tag count for the subsequent round is almost the same as the actual tag count.

The numerical results, a sample set of which is given in Table 2, consistently suggest that the average of the tag count estimate for the alternative method compares very closely with the average of ML estimator, even for smaller values of N and K . Note that the alternative tag count estimation method does not consider the observations Z_S and Z_U . The ML tag estimation algorithm cannot be invoked when $Z_I = 0$. Similarly, the alternative estimation method cannot be used when $Z_I = 0$ or 1. In such cases, the tag count is adjusted as the lower bound ($Z_S + 2Z_U$). This is the reason behind the more significant error in the tag count estimate for $N = 25$ and $K = 80$ in Table 2. In all other scenarios, the average of the tag count estimate for both methods is very close to the actual tag count.

3.3. Comparison with previous work

In ASAP and the frame size adaptive MAC protocols in [11–14], tag count estimation is performed by using the available

² We assume that the reader prematurely closes the slot if there is no response after 10 bits, which leads to 40 microseconds of idle slot duration.

TABLE 2: Comparison of estimation methods.

Actual tag count (K)⇒ Slots (N) ↓	10				80			
	ML	Exp	[12]	[14]	ML	Exp	[12]	[14]
25	10.184	10.1	11.167	10.4285	62.44	62.44	78.0308	52.5408
50	9.969	9.771	11.14	10.4361	81.66	81.78	78.6	73.3524
100	9.989	9.932	10.78	10.4442	* ^a	80.584	78.43	81.7970

^(a) ML estimator is infeasible.

information at the reader. In [11], the tag count is estimated by the simple lower bound ($Z_S + 2Z_U$). Although this estimate is simple, it may not be accurate. Given this estimate of the tag count, the protocol in [11] calculates an optimum frame size as well as its corresponding read cycle, which is the maximum number of rounds the reader performs with the current frame size. These values are obtained by minimizing the reading time with a particular probability of reading all tags. These values are computed and saved as a look-up table at the reader. In [12, 13], tag count estimation is based on finding the probability distribution of the number of tags transmitting. The estimated number of tags is used to find the optimum frame size maximizing the expected throughput of framed-ALOHA. This optimization yields that the optimum frame size is equal to the estimated number of tags. This protocol is shown to outperform the protocol in [11] in terms of tag estimation [12]. In [14], the tag count is estimated by using the simple equation ($Z_S + \alpha Z_U$). The constant, α , is set to be 2.39 [24]. The frame size for passive tags is given as the following relation [14]:

$$N = H * K, \quad (11)$$

where $H \in [1, 1.4]$ and K is the estimated tag count. While for our ASAP, the reader requires knowledge of the optimum multiplier only, the RFID reader employing in [11] requires a look-up table, which contains the optimum frame size and the corresponding number of read cycles. In addition, since initial tag count is generally not available at the reader, obtaining the exact size of the look-up table is not possible. Thus, the reader must maintain a large size of the look-up table which leads to an increase in the memory requirement at the reader.

The protocol in [11] can have potentially high complexity for calculating the look-up table for a large number of tags. This complexity stems from the calculation of factorial operations. Similarly, the protocol in [12, 13] can also lead to high complexity for estimating the tag count which results from the involved factorial operation to calculate probability distribution. ASAP bypasses such computationally expensive operations by using the simpler estimate in (10) which also requires the information of idle slot count only.

Lastly, the protocol in [11] is limited to static RFID systems, where the same tags stay in the reader's field indefinitely. In the dynamic scenario where tag population can be dynamically changed, the notion of read cycle in [11] (which results in the repeated operation of the same frame size) may not lead to good performance.

Table 2 shows the performance of tag count estimation of ASAP and other existing protocols discussed in the section. We observe that the simple estimation algorithm of ASAP performs almost equally well and sometimes even better than the protocol in [12]. For large tags with small initial number of frames, we observe that the protocol in [12] estimates tag count better. The simple protocol in [14] also performs quite well. However, the estimation is not quite accurate for large tags with small number of initial frame sizes.

3.4. Adaptation of ASAP on Class-1 Generation-2 RFID MAC protocol

The MAC protocol of Class-1 Generation-2 (c1gen2) RFID system is also based on time-slotted ALOHA and communications between the reader and the tags take place in inventory round [15]. Each inventory round consists of number of slots and the size of the round can vary. However, c1gen2 does not attempt to estimate the tag count. At the start of each inventory round, the reader broadcasts "Query" command and the command contains the slot count parameter (Q). Q is any integer value between 0 and 15 and determines the size of the round (2^Q). The Q selection algorithm is defined in [15]; Q is increased by a constant C whenever a collision occurs and decreased by C whenever an idle slot is detected. Successful slots do not change Q . C is given by $0.1 \leq C \leq 0.5$ in [15]. Upon receiving the Query command, the tags randomly select a number in the range $(0, 2^Q - 1)$ and store the number in their slot counter. The Q selection algorithm is simple, but there is no notion of finding the optimum Q . Thus, it is clear that the performance of the MAC protocol of c1gen2 is affected by the choice of Q . A small value of Q for a large population of tags results in unacceptable many collisions. A large value of Q for a small population of tags results in waste of time-slots. The design of an algorithm to find the appropriate value of Q given the population of tags in the reader field is therefore important. However, the algorithm for choosing the value of Q is not specified in the standard, and is left open for implementers. Thus, the estimation algorithm of ASAP can be directly implemented on the MAC of c1gen2 for choosing the slot count parameter in each round.

4. THE EXPECTED TOTAL READ TIME

In this section, we derive the expressions for the expected total identification time for reading K tags. The reader recursively offers rounds with adaptive frame sizes $N_j = \beta K_j$, where N_j is the frame size in the j th round and K_j is the

unidentified tag count at the beginning of the j th round. Define T_j as the expected time duration of the j th round. Then

$$T_j = T_B E[S_j] + T_B E[U_j] + T_I E[I_j], \quad (12)$$

where S_j , I_j , and U_j denote the successful, idle, and unsuccessful slots, respectively, in the j th round. For large K_j , (12) simplifies to

$$T_j = T_B \beta K_j [1 - (1 - \alpha)e^{-1/\beta}]. \quad (13)$$

We define total expected identification time T as

$$T = \sum_{j=1}^{\infty} T_j = T_B \beta [1 - (1 - \alpha)e^{-1/\beta}] \sum_{j=1}^{\infty} K_j, \quad (14)$$

where K_j is given by

$$K_j = K_{j-1} - E[S_{j-1}] = K_{j-1}(1 - e^{-1/\beta}). \quad (15)$$

Using (14) and (5), we get

$$T = T_B \beta [1 - (1 - \alpha)e^{-1/\beta}] K e^{1/\beta} \quad (16)$$

which is the total identification time of $K_1 = K$ tags when the reader employs the policy of offering an adaptive frame size given by β times the number of identified tags participating in the round. We note that using (5) for derivation of T results in an underestimate of the actual duration of a round for small K because $(1 - (1/\beta K))^K$ is smaller than $e^{-1/\beta}$ for small K , thus, this analysis can be considered a pessimistic view of the performance of the proposed policy.

5. p -ASAP

Until now, we did not impose any constraint on the frame size. We allowed it to increase arbitrarily, as a function of the tag population. In practice, tracking the number of idle slots within a large frame could become cumbersome. A large frame size also increases the wait time for an unsuccessful tag, since a tag is allowed only one transmission in a frame. Further, in factory production setups, tags attached to manufactured parts and produced commodities arrive into an RFID field and depart after remaining in the field for some fixed time, owing to the motion of conveyor belt or otherwise. These setups imply a time constrained presence of RFID tags and the challenge is to identify these tags before they depart. Because of these constraints, the reader may have to expedite the transmissions by these tags. Consequently, the long wait time for an unsuccessful and time-critical tag in these dense, mobile RFID systems is definitely not acceptable. To cater these, we extend the design of ASAP to scenarios with a constrained frame size by introducing the p -ASAP (where p stands for the round access probability).

In p -ASAP, the reader broadcasts an additional parameter, called the “round selection probability” in the “reader command.” The purpose of this parameter is to request each tag to first choose to participate in the round with probability, p . If the result of the random experiment is favorable, then the tag proceeds as in ASAP, that is, chooses a slot in the

frame and schedules the transmission of the EPC string. If unfavorable, the tag transitions back to the “activated” state. In effect, this parameter reduces the average “active” tag population in the round, in view of the frame size constraint. We set the length of the “round selection probability” field to 4 bits long which can represent up to 16 levels of p . Empirically, we observe that this yields a sufficiently small quantization error on p . The tag state machine also requires modifications to support the transition from the “select and transmit” state to the “activated” state in the event of an unfavorable result. The basic functioning, the system model, and the other assumptions remain the same as in ASAP.

In p -ASAP, the effective probability of selecting a slot changes to $p_e = p/N$. Consequently, the expected count of successful, idle, and unsuccessful slots are modified as

$$\begin{aligned} E[S] &= pK \left(1 - \frac{p}{N}\right)^{K-1}, & E[I] &= N \left(1 - \frac{p}{N}\right)^K, \\ E[U] &= N - pK \left(1 - \frac{p}{N}\right)^{K-1} - N \left(1 - \frac{p}{N}\right)^K. \end{aligned} \quad (17)$$

Using approximation for large numbers defined in (5), the efficiency function is given by

$$p^{\text{eff}} = \frac{1}{\phi e^{(1/\phi)} + (\alpha - 1)\phi}, \quad (18)$$

where $\phi = \beta_{p\text{-ASAP}}/p$ and the optimum $\phi^* = 1.943$, or equivalently the optimum $\beta_{p\text{-ASAP}}^* = 1.943p$. We note that ML estimator of p -ASAP requires a modification of that of ASAP. That is, in p -ASAP, the reader should include the tags that transitioned back to the “activated” state by dividing the estimated tag count of ML estimator of ASAP with p , that is, $\hat{K} = \hat{K}_{\text{ML}}/p$ to get the desired tag count estimate. Similarly, before invoking the frame size decision algorithm for the next round, the reader must exclude the tags that are going to be transitioned back to the “activated” state in the next round by multiplying p . Hence, we propose the frame size in j th round as follows:

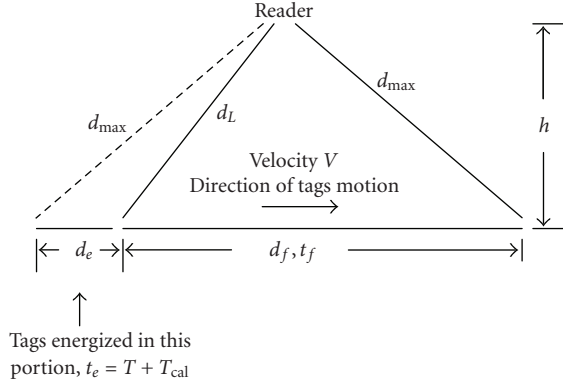
$$N_j = 1.943 \left(\frac{\hat{K}_{\text{ML},j-1}}{p} - Z_{S_{j-1}} \right) p. \quad (19)$$

When the reader offers an appropriate frame size with round selection probability, p for K_j tag, then expected duration of a round can be computed as

$$T_j = T_B p \beta K_j [1 - (1 - \alpha)e^{-1/\beta}]. \quad (20)$$

As expected, the average duration of a round is less than that of ASAP. However, the expected total time for identifying K tags is found to be the same as that of ASAP. Since reducing the slot access probability does not impact the expected total time, the decrease in the expected duration of a round is compensated by the increase in the number of rounds.

The parameter p should be chosen in accordance with the frame size constraint on the system. Denote the maximum number of slots by N_{max} , given K tags, the parameter that yields the optimal throughput in the round is chosen to be $p = \min(N_{\text{max}}/\beta K, 1)$. Thus, when the number of

FIGURE 3: *m*-ASAP system model.

tags to be identified is small, we need to revert back to original ASAP. Basically, the reader first computes the frame size $N_i = \beta K_i$ and if $N_i > N_{\max}$, then it computes the round selection probability p that reduces the expected “active” tag count to a value that satisfies the equation

$$N_{\max} = p\beta K_i. \quad (21)$$

If $N_i \leq N_{\max}$, the reader is not required to calculate the value of “round selection probability.” Essentially, the reader offers rounds of variable frame size which are limited by N_{\max} . In the process, it may offer a variable “round selection probability” calculated in view of the unidentified tag count in each round.

6. *m*-ASAP

The biggest challenge that the mobile tags introduce is the time-constrained presence in the RFID field. The time the tag will spend in the reader’s field clearly depends on the speed and the coverage of the reader. Tag density in the reader’s field also affects the performance. In mobile RFID systems, the tag’s basic communication mechanism is still the same: a tag enters the field, collects frame size information, and repeatedly attempts the transmission of identification string. The difference is that the tag mutes not only when its transmission succeeds but also when it departs from the RFID field, whichever occurs first. Also, in this setup, new tags continuously arrive into the RFID field. Consequently, a substantial tag population is there to schedule the transmissions in every round. We propose *mobile (m)*-ASAP for such RFID system setups and we focus on a design that improves the percentage of identified tags in the backdrop of the restricted time-presence of RFID tags. In particular, we concentrate on the dual of the problem of finding the read performance of a particular arrival model and consider the design of the initial tag count, the tag arrival rate, and the tag departure rate in a mobile RFID system, such that $P\%$ tags are identified. Hence, the percentage requirement for the read performance serves as the QoS requirement for our system.

We assume that passive tags arrive into an RFID reader’s field on a conveyor belt moving at a constant velocity, V . The system model for *m*-ASAP is shown in Figure 3. In a stationary RFID system, the reader schedules the transmission of the “reset” and “oscillator calibration signal” cycle in the beginning of an identification process to energize and synchronize the tag’s IC chip. In the mobile setting, however, new tags arrive in the middle of an identification process and hence we need the additional intermediate “oscillator calibration signal” cycle to provide the synchronization information. Therefore, we propose that in *m*-ASAP, the reader schedules the “oscillator calibration signal” cycle of duration T_{cal} before the beginning of every new round as shown in Figure 4. The combination of the “oscillator calibration signal” cycle, followed by a “Null” and a “round” is defined as the extended round of duration T .

We denote the maximum operating range of the reader as d_{\max} and the vertical distance between the reader and the conveyor belt as h . We denote the total time spent by each tag in the RFID field as $t = t_e + t_f$. Here, t_e is the time during which new arriving tags energize and collect synchronization information and t_f is the time during which the tags schedule the transmission of their packets, that is, EPC and CRC. We choose $t_e = T + T_{\text{cal}}$ as it ensures that new tags receive at least one “calibration cycle” after collecting sufficient power while they transit the distance d_e . Accordingly, we compute t_f as $t_f = (2\sqrt{d_{\max}^2 - h^2}/V) - t_e$. The new tags enter the zone d_f when the reader broadcasts an intermediate “reader command” and this instance also marks the beginning of each tag’s infield timer. We denote the tags that enter the reader’s field at the stroke of the i th “reader command,” or equivalently at the beginning of the round i as group G_i tags. Therefore, the timer t_f for a group of tags arriving together in the reader field will expire at the same time.

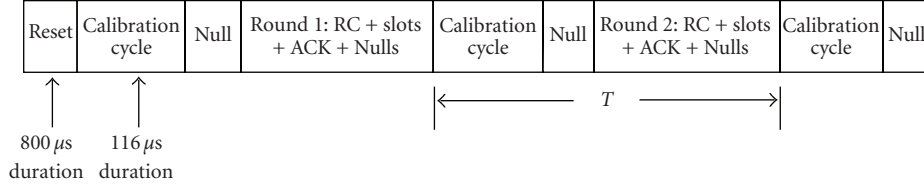
We define the tag arrival rate as ψ . Since the tags are moving within the reader’s field at a constant velocity, the tag arrival rate is equal to the tag departure rate. Other assumptions in the system model remain the same as before.

In the mobile scenario, we design the MAC such that $P\%$ of tags are identified. This will be accomplished by offering a sufficient number of rounds within time t_f for tags arriving in each group such that the desired percentage of the tags from each group are identified.

Assume that G_1 tags arrive into the reader’s field at the beginning of the first round. By design, ASAP will dictate that the reader offers a frame size of $N = \beta G_1$ to optimize the efficiency of the first round. Recall that in this case, the expected time of a round is given by (13). Thus, in *m*-ASAP, we have

$$T = T_B \beta G_1 [1 - (1 - \alpha)e^{-1/\beta}] + T_{\text{overhead}}, \quad (22)$$

where $T_{\text{overhead}} = T_{\text{Cal}} + T_{\text{Null}} + T_{\text{RC}} + T_{\text{Ack}} = 403.5 \mu\text{s}$ consistent with the suggested standards [3]. The key in *m*-ASAP is to keep an approximate constant number of tags in each round (G_1) leading to a duration of T per round. This in turn dictates an arrival rate ψ that can guarantee $P\%$ tag identification.

FIGURE 4: m -ASAP round structure.

The desired arrival rate can be found as follows. For large values of G_1 and N , the expected number of successful tags is given by

$$E[S_1] = G_1 \left(1 - \frac{1}{N}\right)^{G_1-1} = G_1 e^{-1/\beta}. \quad (23)$$

We thus require the number of new tags that arrive in the second round as $G_2 = G_1 e^{-1/\beta}$. When ψ is the tag arrival rate, then the expected value of new tags in the round will be given by ψT . Therefore, ψ must satisfy $\psi T = G_1 e^{-1/\beta}$:

$$\psi = \frac{G_1 e^{-1/\beta}}{T_B G_1 \beta [1 - (1 - \alpha) e^{-1/\beta}] + T_{\text{overhead}}}. \quad (24)$$

In order to find G_1 , we take advantage of (15), which gives us the percentage of unidentified tags left when the reader recursively offers n rounds of appropriate frame sizes in ASAP. Recall that

$$K_n = K_1 (1 - e^{-1/\beta})^{(n-1)}. \quad (25)$$

Equivalently, the percentage of tags that remain at the beginning of the n th round is K_n/K_1 100(%). Basically, if the reader offers an appropriate frame size in every round in view of the instantaneous tag population, then for large number of experiments, the total number of offered slots in each round will divide proportionally to the remaining tags of each group. In view of this, we can separate the tags from each group and can perform an independent analysis on each group of tags. Hence, we use (25) to find the number of rounds, n_r that every group of tags must participate in, such that the individual percentage of tags identified from every group is $P\%$. We obtain $n_r = \lceil (n - 1) \rceil$ from

$$\frac{100 - P\%}{100} = (1 - e^{-1/\beta})^{(n-1)}. \quad (26)$$

Subsequently, we use n_r to compute the acceptable expected round duration as $T = (t - t_e)/n_r = (t - T_{\text{cal}})/(n_r + 1)$. We substitute T in (22) to compute G_1 as

$$G_1 = \frac{(t - T_{\text{cal}})/(n_r + 1) - T_{\text{overhead}}}{T_B \beta [1 - (1 - \alpha) e^{-1/\beta}]}. \quad (27)$$

We further substitute the value of G_1 in (24) to compute the arrival rate that should be met for the target $P\%$.

Note that in this design, the reader attempts to offer an approximately fixed duration frame in every round. However, each tag chooses a slot randomly and independently and

we also know that the duration of an idle slot is different from the duration of a busy slot within a frame. Consequently, the deviation in the statistics (Z_I, Z_S, Z_U) in a given frame has the effect of producing a variable duration round. In view of this discussion, it is possible that the timer of a particular group of tags may expire before they participate in all n_r rounds. Hence, we propose that the reader should design for either $\lceil (n - 1) \rceil + 1$ rounds, or use a threshold time, $t_{\text{th}} < t_f$ to compute T and G_1 .

7. ASAP IN THE PRESENCE OF FAULTY TAGS

The passive RFID tags are expected to have simple and inexpensive hardware designs [6]. In view of that, we need to consider the probability that tags may break and not participate despite being present in the RFID field. In other cases, they may not collect sufficient energy to run their microprocessor and other circuitry to decode the reader commands, temporarily. In general, the presence of these tags (faulty tags) impacts the system dynamics and the performance of the RFID systems. In these systems, we address two scenarios: the presence of *physically faulty tags* and the presence of *system faulty tags*.

Physically faulty tags are broken and cannot schedule the transmission of their EPC in any eventuality whatsoever. Quite obviously, these tags will not be identified by the reader. In the setup, we assume that each tag can be physically faulty with a probability p_p and the reader knows the value of p_p . If the reader also has the information about the exact tag count or partial information about the initial distribution, then the reader can use p_p to appropriately propose the frame size. From the tag state machine perspective, these tags will always remain in the “unpowered” state.

Tags are said to be system faulty due to the insufficient accumulation of energy, or temporary loss of synchronization or failure to interpret the contents of the “reader command” appropriately by a particular tag. These tags opt out of the current round by either remaining in the “activated” state or by moving back to the “select and transmit” state, intermediately. We assume that each unidentified and system faulty tag drops out of a round with a probability, p_s . Unlike the physically faulty tags, the system faulty tags can participate in the next round if they can accumulate sufficient energy or resolve their synchronization problems.

The presence of these faulty tags prompts a modification on the ML estimator and effects on the frame size to be chosen. We consider that the reader knows p_p and p_s in

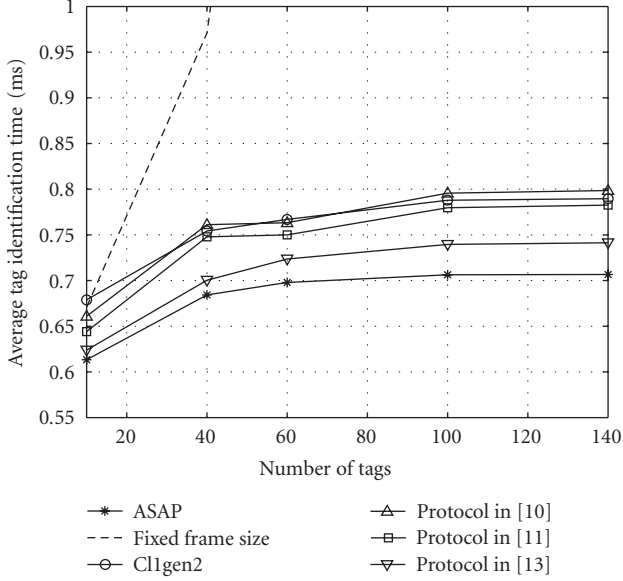


FIGURE 5: ASAP versus protocols in [11, 12, 14, 15] versus fixed frame size: average tag identification time.

advance³, however it does not have any information of the tag count that actually participate. In that case, since the reader's estimate of the tag count is based on its observations of the activity in the round, the value of p_p is irrelevant. Further, since the collected observations (Z_I , Z_S , and Z_U) correspond to tags that actually participated in a round, the ML estimation algorithm does not provide any information about the existence of system faulty tags. Thus, the reader should make an adjustment for the appropriate tag count estimate by dividing the estimated tag count of ML estimator with p_s , that is, $\hat{K} = \hat{K}_{ML}/(1 - p_s)$.

Similarly, before invoking the frame size decision algorithm, the reader must exclude system faulty tags (by multiplying $(1 - p_s)$), which may not participate in the next round owing to temporary faultiness. Hence, we propose the frame size in j th round as follows:

$$N_j = 1.943 \left(\frac{\hat{K}_{MLj-1}}{(1 - p_s)} - Z_{Sj-1} \right) (1 - p_s). \quad (28)$$

8. NUMERICAL RESULTS

In this section, we provide our simulation results of the performance of the proposed protocols. We simulate the following results by using MATLAB. We assume the 64-bit EPC and 16-bit CRC data structure. Thus, the optimum β^* is set to be 1.943 in the sequel. We focus on the average tag count identification time and demonstrate the performance of ASAP. The average tag count identification time is the total identification time divided by the total number of tags. The proper

number of slots are adaptively proposed in each round, based on the estimate (given by (9) and (10)) of the number of tags identified. The simulation ends when all tags are identified and total number of rounds and the corresponding round size are noted. As shown in Figure 1, each round consists of a number of sequence data slots and overhead slots (null, ACK command, and reader command). We do not consider the processing time for tag count estimate and data transmission time.

In Figure 5, we compare the average tag count identification time for ASAP, other existing protocols [11, 12, 14, 15], as well as fixed frame size where the reader offers the same frame size for every round. In order to ensure a fair comparison, the initial frame size for all protocols is selected to be 16 consistent with the protocol in [11]. For the protocol in [11], the probability of identifying all tags is set to be 0.99. For the protocol in [14], H in (11) is set to be 1.4. For the Q selection algorithm of Class-1 Generation-2, the initial Q is set to be 4 and C is chosen as $0.8/Q$ [12].

We observe that ASAP outperforms all other protocols owing to either the more accurate tag count estimation or the corresponding frame size adjustment. As noted in Table 2, although the tag count estimation in [12, 14] is as accurate as that of ASAP, we observe that ASAP performs better. This shows the feasibility and advantage of the optimum frame size adjustment of ASAP. We expect that the performance benefit of ASAP might be even more pronounced if the processing time for tag count estimation is considered due to its computationally simple estimation algorithm.

In addition, the frame size adaptive protocols including ASAP perform better than the fixed frame size as expected. This shows a clear advantage of the frame size adaptive MAC protocols versus the fixed frame size protocol. The average reading time was obtained for relatively small number of tags, that is, up to 140 tags. This is because for a large number of tags, the look-up table of the protocol in [11] and probability distribution for the number of tags in [12] is prohibitively complex to obtain. Convinced by the performance advantage of ASAP over these protocols, in the sequel, we provide further simulation results of ASAP under a wide range of tag populations and scenario.

In Figure 6, the average tag identification time (T_{Av}) for ideal ASAP, that is, the reader has the exact tag count, is significantly better than the fixed frame policies for any K and is approximately constant. In contrast, the fixed frame size policies show the best results when $N \approx 2K_1$. When the frame size offsets by a large value from $N \approx 2K_1$, the T_{Av} increases rapidly. For example, a very high values of T_{Av} was observed when (i) $K_1 \geq 200$ with $N = 50$, and (ii) $K_1 \geq 500$ with $N = 100$. Note that we do not observe the instability problem of ALOHA [7, 25], since the tag count is fixed and it decreases as the successful tags do not transmit in subsequent rounds.

Next, we investigate the performance of ASAP when the reader proposes an arbitrary frame size in the first round and subsequently, it estimates the tag count to propose the optimal frame size. In these simulations, we used the ML estimator, when both N and $Z_I < 80$, and the alternative estimator,

³ Such statistical information is likely to be available from tag manufacturers.

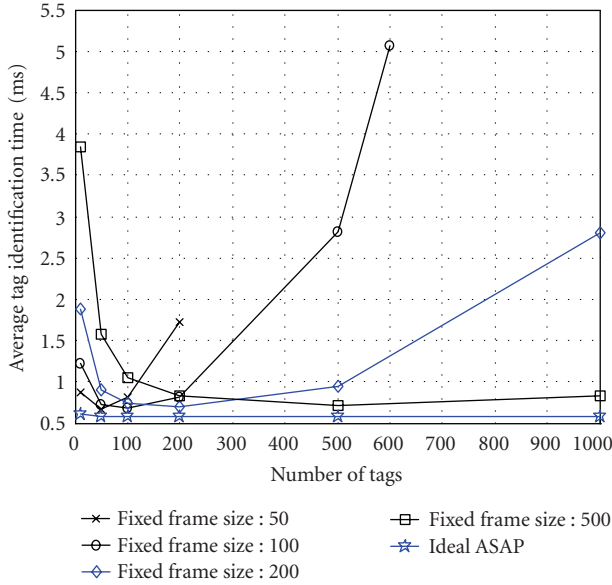


FIGURE 6: Ideal ASAP versus fixed frame size policies: average tag identification time.

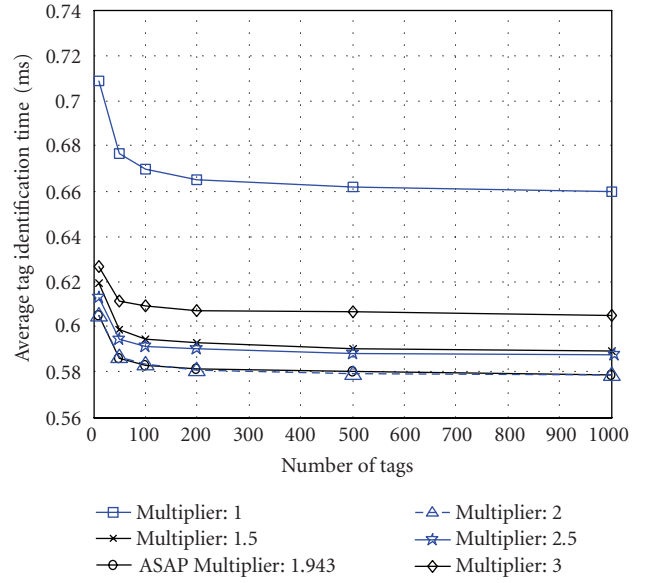


FIGURE 8: Performance of $N = \beta K$ -type policies.

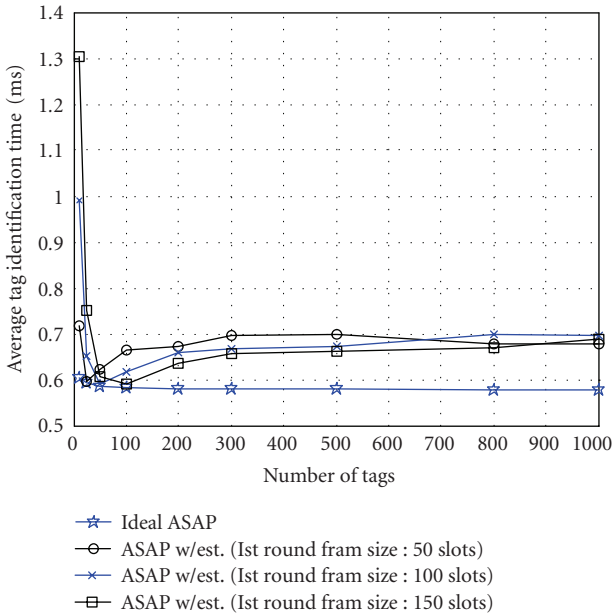


FIGURE 7: Ideal ASAP versus ASAP: average tag identification time.

otherwise. In Figure 7, We observe that T_{Av} remains below 0.7 millisecond for any large K , even when the frame size is small in the first round. This choice of frame size, however, impacts the T_{Av} for small tag count significantly, for example, T_{Av} of 1.3 milliseconds, when K is 10 and $N_1 = 150$.

Figure 8 compares the performance of ASAP with different multipliers. As expected $\beta^* = 1.943$ performs the best. We observe that the multiplier values close to the optimum value, for example, 2, perform almost as well.

In Figure 9, we show the performance of p -ASAP when the frame size is limited. We assume that the reader does not

have any prior information about the actual tag count. The reader begins the identification process for every K with a frame size equal to the maximum frame size and a round selection probability of 1. Subsequently, the reader estimates the tag count and takes into consideration N_{max} , to offer the frame size and round selection probability, p . We consider maximum frame sizes (N_{max}) of 50, 100, 200, and 500. We observe that the average tag identification time is small and p -ASAP performs well in most of the cases except for the case where K is small and N_{max} is large ($K = 50$ and $N_{max} = N_1 = 500$) or K is large and N_{max} is small ($K = 1000$ and $N_{max} = N_1 = 50$). First case is due to the waste of slots. Second case is due to the large number of tags with small size of frames.

For m -ASAP, we performed sets of simulations for QoS requirement of $P_{\%} = 99\%$ and 99.9% , respectively. We set $V = 5$ m/s, $h = 1$ m, and $d_{max} = 2$ m to get $t = 692.82$ milliseconds. The exit criterion of each iteration is the arrival of a total of 50000 tags in the reader's field. The tags arrive according to a Poisson distribution with the arrival rate ψ , that is, determined for one target $P_{\%}$. The results are given in Table 3. We observe that m -ASAP shows impressive performance in terms of the achieved percentage. We also notice the improvements, when we offer an additional round to each group of tags to ensure that each group of tags must participate in at least n_r rounds. In both set of simulations, we observe that T_{Av} remains close to 0.58 millisecond. This is because the design of m -ASAP ensures that an appropriate frame size is offered in each round.

In Figure 10, we show the performance of the average tag identification time T_{Av} for different combinations of p_s and p_p . The reader arbitrarily offers the first frame size, $N_1 = 100$, when the number of tags, K , is 1000. We observe that as p_s is increased for fixed p_p , the T_{Av} improves slightly. The performance deteriorates, although not significantly, as the difference between the optimal and offered frame size increases.

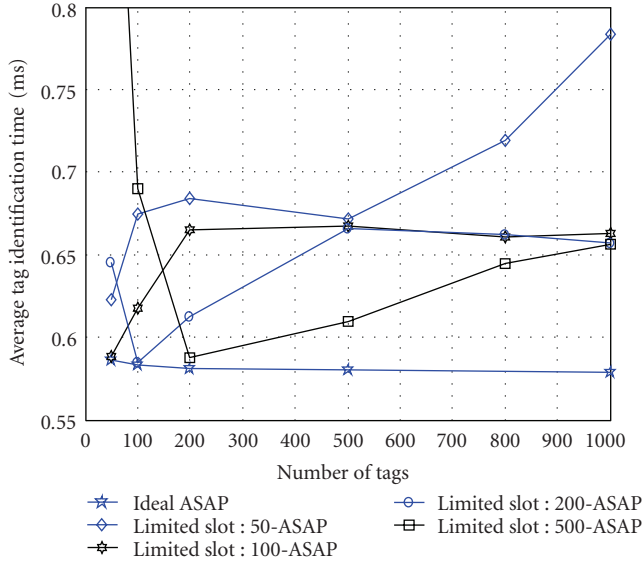
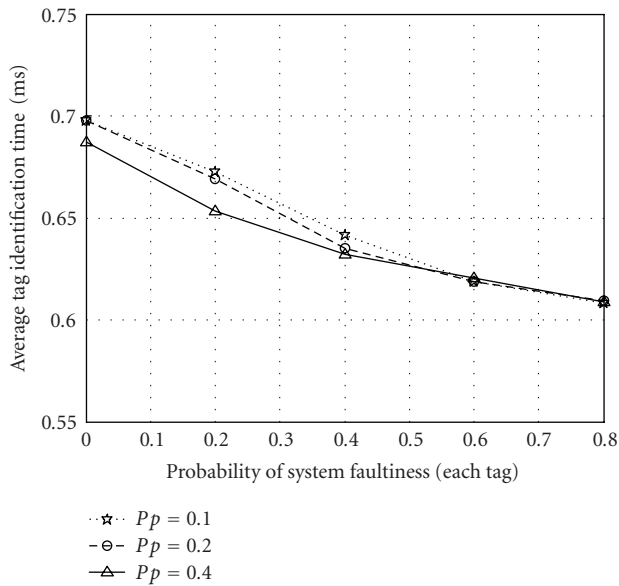


FIGURE 9: Performance of ASAP with limited frame size.

FIGURE 10: Trend for the average tag identification time for different combinations of p_p and p_s .

This fact provides a rationale for the trend in these simulations. When p_s is high, the expected number of participating tags in the first round reduces to a small count. Since the reader offers a frame size of 100 slots in the first frame, the increase in p_s has the effect of improving the performance in the first round because N_1 and $K_{1,\text{participate}}$ closely follow the relation $N_1 \approx \beta K_{1,\text{participate}}$ as against the other choices of p_s . The choice of p_p does not effect the average tag identification time.

In another set of simulations, N_1 is increased from 100 to 200 and 500 slots, respectively, and the average tag identification time is shown in Figure 11. We keep the same tag count ($K = 1000$ tags) and $p_p = 0.1$. In ASAP, $N_1 = 500$ is

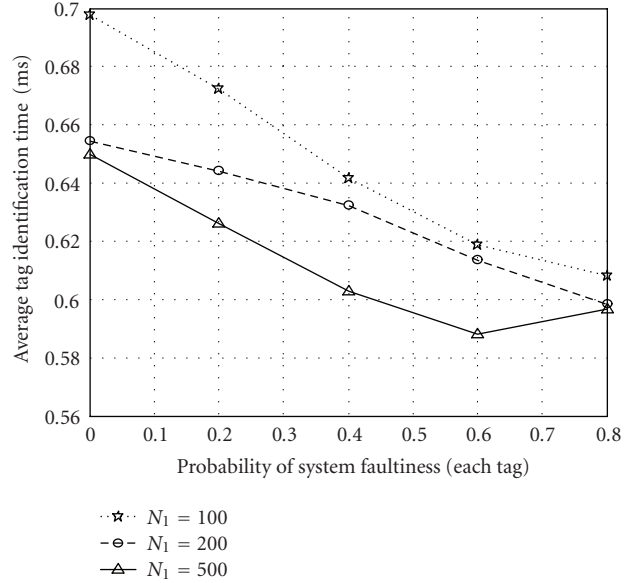


FIGURE 11: Trend for the average tag identification time, when the first round frame size is varied.

TABLE 3: Performance results: m -ASAP with tag count estimation.

Target $P_\%$	Extra round	n_r	G_1	N_1	Achieved $P_\%$
99	X	6	285	553	99.85
99	✓	7	249	484	99.92
99.9	X	8	221	430	99.96
99.9	✓	9	199	387	99.98

optimal, if the participated tag count is approximately close to $N_1/\beta = 258$ tags. When $p_s = 0.6$ and 0.8 , the expected tag count in the first round reduces to 360 and 180, respectively, as against a higher expected tag count for smaller p_s . As a result, T_{AV} is lower for higher p_s . Similar explanations hold good for $N_1 = 100$ and 200 .

We note that ASAP with $p_s = 0$ is essentially the same as ASAP with estimation and for reasons discussed above, we observe that performance in realistic ASAP improves if the probability p_s is such that it brings the expected tag count to a desirable value in view of the frame size of the first round.

9. CONCLUSIONS

In this paper, we have proposed ASAP, a MAC protocol tailored for RFID systems with passive tags. Specifically, ASAP takes advantage of the fact that the envisioned RFID systems with passive tags will be collision limited, and utilizes tag count related information inherently collected at the RFID reader to adjust the frame size in a framed slotted ALOHA setting. The MAC protocol relies on obtaining an estimate of the number of tags in the reader's field based on the observation of the number of idle, unsuccessful, and successful slots in the current frame, to determine the size of the next frame. It is shown that the proposed adaptive MAC protocol improves the currently suggested slotted ALOHA-based STAC as well as previously suggested protocols significantly

in terms of average tag identification time of the tags. We have also extended the design of ASAP to variants of ASAP, that is, p -ASAP, m -ASAP, and ASAP with faulty tags. For each scenario, we have modified the ML estimator appropriately for providing the best performance. We have shown that the performance of these ASAP is also impressive.

The protocol proposed in this paper aims to gain significant performance improvement with virtually no additional complexity over existing standards. To that end, we note that the frame size can be further fine tuned by assuming estimators with memory at the expense of additional complexity.

APPENDIX

A. PROOF OF STRICT PSEUDOCONCAVITY OF THE EFFICIENCY FUNCTION

We provide the proof that the efficiency function defined in (6) is a strictly pseudoconcave function and hence the local maximum ($\beta = 1.943$) found for this function is also the global maximum value. We have

$$p_{\text{eff}} = \frac{1}{\beta e^{1/\beta} + (\alpha - 1)\beta} = f(\beta). \quad (\text{A.1})$$

A differentiable function $f(\beta)$ is strictly pseudoconcave [22] if for $\beta_1 \neq \beta_2$,

$$f(\beta_1) \geq f(\beta_2) \quad \text{implies} \quad (\beta_1 - \beta_2) \frac{df(\beta_2)}{d\beta_2} > 0. \quad (\text{A.2})$$

In other words, if

$$\frac{1}{\beta_1 e^{1/\beta_1} + (\alpha - 1)\beta_1} \geq \frac{1}{\beta_2 e^{1/\beta_2} + (\alpha - 1)\beta_2}, \quad (\text{A.3})$$

then the following equation must be satisfied for strict pseudoconcavity:

$$T(\beta_1, \beta_2) = (\beta_1 - \beta_2)(1 - \alpha) + e^{1/\beta_2} \frac{(1 - \beta_2)}{\beta_2} (\beta_1 - \beta_2) > 0. \quad (\text{A.4})$$

Rearranging (A.3), we obtain

$$(1 - \alpha)(\beta_1 - \beta_2) + e^{1/\beta_2} (\beta_2 - \beta_1 e^{(1/\beta_1) - (1/\beta_2)}) \geq 0. \quad (\text{A.5})$$

By using the Taylor series expansion of the exponential function, (A.5) is rewritten as

$$(1 - \alpha)(\beta_1 - \beta_2) + e^{1/\beta_2} \left[\beta_2 - \beta_1 \left(1 + \left(\frac{1}{\beta_1} - \frac{1}{\beta_2} \right) + \frac{1}{2!} \left(\frac{1}{\beta_1} - \frac{1}{\beta_2} \right)^2 + \dots \right) \right] \geq 0. \quad (\text{A.6})$$

We can decompose (A.6) into two parts as follows:

$$T(\beta_1, \beta_2) - \left(\beta_1 e^{1/\beta_2} \left[\frac{1}{2!} \left(\frac{1}{\beta_1} - \frac{1}{\beta_2} \right)^2 + \frac{1}{3!} \left(\frac{1}{\beta_1} - \frac{1}{\beta_2} \right)^3 + \dots \right] \right) \geq 0. \quad (\text{A.7})$$

With $T(\beta_1, \beta_2)$ given in (A.4) and depending on the sign of $x = ((1/\beta_1) - (1/\beta_2))$, the rest part of (A.7) is either one of the following equations.

$$\begin{aligned} \beta_1 e^{1/\beta_2} \left[\frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \frac{1}{4!} x^4 \dots \right] &= \beta_1 e^{1/\beta_2} (e^x - 1 - x) \quad \text{for } x > 0 \\ \beta_1 e^{1/\beta_2} \left[\frac{1}{2!} x^2 - \frac{1}{3!} x^3 + \frac{1}{4!} x^4 \dots \right] &= \beta_1 e^{1/\beta_2} (e^{-x} - 1 + x) \quad \text{for } x < 0, \end{aligned} \quad (\text{A.8})$$

where both $e^x - 1 - x$ and $e^{-x} - 1 + x$ are strictly larger than zero for all $x \neq 0$. They are zero for $x = 0$. We note that $x = 0$ corresponds to $\beta_1 = \beta_2$. Consequently, from (A.7), we note that $T(\beta_1, \beta_2)$ must be strictly larger than zero for all $\beta_1 \neq \beta_2$. Hence, we proved that the efficiency function is strictly pseudoconcave.

ACKNOWLEDGMENTS

This paper is supported in part by Techcollaborative Round 11 project "Design of Efficient RFID Systems." This paper was presented in part at IEEE International Conference on Communications ICC'06, June 2006, Istanbul, Turkey.

REFERENCES

- [1] J. R. Tuttle, "Traditional and emerging technologies and applications in the radio frequency identification (RFID) industry," in *Proceedings of the IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 5–8, Denver, Colo, USA, June 1997.
- [2] R. Want, "Enabling ubiquitous sensing with RFID," *Computer*, vol. 37, no. 4, pp. 84–86, 2004.
- [3] "13.56 MHz ISM band class 1 radio frequency identification tag interface specification: recommended standard, version 1.0.0," Tech. Rep., Auto-ID Center, Cambridge, Mass, USA, May 2003.
- [4] "Draft protocol specification for a 900 MHz class 0 radio frequency identification tag," Tech. Rep., Auto-ID Center, Cambridge, Mass, USA, February 2003.
- [5] R. Weinstein, "RFID: a technical overview and its application to the enterprise," *IT Professional*, vol. 7, no. 3, pp. 27–33, 2005.
- [6] R. Glidden, C. Bockorick, S. Cooper, et al., "Design of ultra-low-cost UHF RFID tags for supply chain applications," *IEEE Communications Magazine*, vol. 42, no. 8, pp. 140–151, 2004.
- [7] L. Kleinrock and S. Lam, "Packet switching in a multiaccess broadcast channel: performance evaluation," *IEEE Transactions on Communications*, vol. 23, no. 4, pp. 410–423, 1975.
- [8] S. Lam and L. Kleinrock, "Packet switching in a multiaccess broadcast channel: dynamic control procedures," *IEEE Transactions on Communications*, vol. 23, no. 9, pp. 891–904, 1975.
- [9] "860MHz—930MHz class-1 radio frequency identification tag radio frequency identification tag protocol specification candidate recommendation, version 1.0.0," Tech. Rep., Auto-ID Center, Cambridge, Mass, USA, June 2003.
- [10] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '00)*, pp. 75–84, Boston, Mass, USA, August 2000.

- [11] H. Vogt, "Multiple object identification with passive RFID tags," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02)*, vol. 3, pp. 651–656, Hammamet, Tunisia, October 2002.
- [12] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '06)*, pp. 457–462, Pisa, Italy, March 2006.
- [13] C. Floerkemeier and M. Wille, "Comparison of transmission schemes for framed ALOHA based RFID protocols," in *International Symposium on Applications and the Internet Workshops (SAINT '06)*, pp. 92–95, Phoenix, Ariz, USA, January 2006.
- [14] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Transactions on Communications*, vol. E88-B, no. 3, pp. 991–999, 2005.
- [15] "Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz—960 MHz version 1.0.9," Tech. Rep., EPCglobal, January 2005.
- [16] R. Rivest, "Network control by Bayesian broadcast," *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 323–328, 1987.
- [17] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: an anticollision algorithm for the reader collision problem," in *IEEE International Conference on Communications (ICC '03)*, vol. 2, pp. 1206–1210, Anchorage, Alaska, USA, May 2003.
- [18] J. Ho, D. W. Engels, and S. E. Sarma, "HiQ: a hierarchical Q-learning algorithm to solve the reader collision problem," in *International Symposium on Applications and the Internet Workshops (SAINT '06)*, pp. 88–91, Phoenix, Ariz, USA, January 2006.
- [19] J. Wieselthier, A. Ephremides, and L. Michaels, "An exact analysis and performance evaluation of framed ALOHA with capture," *IEEE Transactions on Communications*, vol. 37, no. 2, pp. 125–137, 1989.
- [20] W. Szpankowski, "Packet switching in multiple radio channels: analysis and stability of a random access system," *Computer Networks*, vol. 7, no. 1, pp. 17–26, 1983.
- [21] G. Khandelwal, "Efficient design of dense and time constrained RFID systems," M.S. thesis, The Pennsylvania State University, University Park, Pa, USA, August 2005.
- [22] M. Avriel, W. Diewert, S. Schaible, and I. Zang, *Generalized Concavity*, Plenum Press, New York, NY, USA, 1988.
- [23] S. Ross, *Stochastic Processes*, John Wiley & Sons, New York, NY, USA, 1996.
- [24] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565–568, 1983.
- [25] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Upper Saddle River, NJ, USA, 1992.