A Survey and Tutorial of RFID Anti-Collision Protocols

Dheeraj K. Klair, Kwan-Wu Chin, and Raad Raad

***Abstract*—RFID technologies have revolutionized the asset tracking industry, with applications ranging from automated checkout to monitoring the medication intakes of elderlies. In all these applications, fast, and in some cases energy efficient, tag reading is desirable, especially with increasing tag numbers. In practice, tag reading protocols face many problems. A key one being tag collision, which occurs when multiple tags reply simultaneously to a reader. As a result, an RFID reader experiences low tag reading performance, and wastes valuable energy. Therefore, it is important that RFID application developers are aware of current tag reading protocols. To this end, this paper surveys, classifies, and compares state-of-the-art tag reading protocols. Moreover, it presents research directions for existing and future tag reading protocols.**

***Index Terms*—RFID systems, Anti-collision protocols, Tree variants, Aloha variants, Tag estimation functions.**

I. Introduction

RADIO Frequency IDentification (RFID) systems are becoming ubiquitous. In 2005, over 1.3 billion RFID tags were produced and this figure will rise to 33 billion by 2010 [1]. One of the key factors that drive the growth of RFID is its ability to identify objects wirelessly without line-of-sight. Thus, making RFID particularly attractive for applications in retail, inventory management, and supply-chain management. More recently, with the aim of reducing logistical overheads, costs, and product losses, both Wal-Mart and the Department of Defense have mandated their respective suppliers use RFID tags [2][3][4].

RFID systems consist of a reading device called a reader, and one or more tags. The reader is typically a powerful device with ample memory and computational resources. On the other hand, tags vary significantly in their computational capabilities. They range from dumb passive tags, which respond only at reader commands, to smart active tags, which have an on-board micro-controller, transceiver, memory, and power supply [5]. Among tag types, passive ones are emerging to be a popular choice for large scale deployments due to their low cost [6][7][8].

Collision due to simultaneous tag responses is one of the key issues in RFID systems [7]. It results in wastage of bandwidth, energy, and increases identification delays. To minimize collisions, RFID readers must use an anti-collision protocol. To this end, this paper reviews state-of-the-art tag reading or anti-collision protocols, and provides a detailed comparison of the different approaches used to minimize collisions, and hence help reduce identification delays. Such review will be of great importance to researchers and designers that are building RFID systems involving interrogation zones with varying tag densities – e.g., reading tagged items in a shopping cart quickly as a customer passes an automated checkout. Apart from that, this paper also presents research directions, challenges and problems in RFID systems that use wireless sensors to detect tags.

To date, there are two prior surveys on anti-collision protocols: [9] and [10]. A key limitation of these works is that they only survey protocols published before the year 2004. Other than that, they lack comprehensiveness. For example, the first work, i.e., [9], reviews Aloha variants only. Moreover, it lacks coverage of dynamic FSA (DFSA) protocols, especially those published after the year 2004. In addition, [9] fails to cover DFSA protocols that use a tag estimation function to derive the optimal frame size for use in each round. In the second work, [10] only covers four Aloha and nine tree variants. In contrast, this paper studies 31 variations of tree protocols and 42 Aloha variants. Moreover, we are the first to demonstrate the operation of tree protocols using the same tag set. Apart from that, we also survey five hybrid protocols. Lastly, this paper uses a comprehensive methodology based on firstly identifying the pros and cons of each protocol before presenting an in-depth comparison among various anti-collision schemes. Specifically, we compare these schemes according to their operating principle, system cost, protocol complexity, identification delays, bandwidth requirements, reader or tag hardware requirements, overall performance, and scalability.

The remainder of this paper is organized as follows. We first review RFID technologies and provide a comparison based on their operating principles in Section II. After that we present a comprehensive survey and comparison of Aloha and tree based protocols in Section III-A and III-B respectively. In Section III-C, we survey five hybrid tag reading protocols. This is followed by a review of current RFID standards in Section IV. Next, in Section V, we present issues in emerging RFID systems that involve wireless sensors. Section VI concludes the paper.

II. Background

Before delving into anti-collision or tag reading protocols, we first present how RFID systems operate, and their classifications.

A. Communication Principle

RFID systems communicate using either magnetic or electromagnetic coupling. The difference between these two sys-

Manuscript received 16 June 2008; revised 31 January 2009.
The authors are with the School of Electrical, Computers, and Telecommunications Engineering, University of Wollongong, Northfields Avenue, NSW, Australia (e-mail: {dklair, kwanwu, raad}@uow.edu.au).
Digital Object Identifier 10.1109/SURV.2010.031810.00037

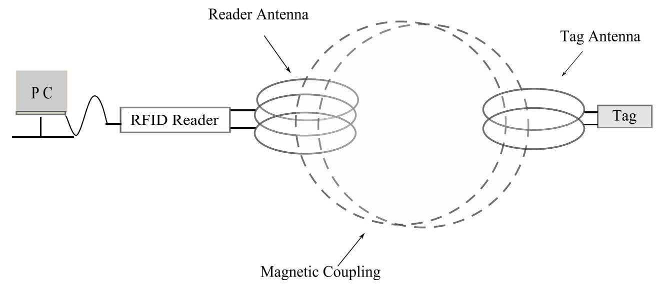


Fig. 1. An inductively coupled RFID system.

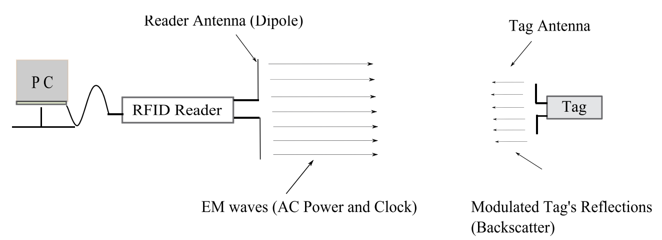


Fig. 2. Backscatter RFID System.

tems lies in their operating field, i.e., near or far field. A key property of far-field communication is that they have a longer read range compared to near field systems. Table I [11][7][5][8] presents a comparison of tags that operate in these fields.

1) Magnetically Coupled Systems: Magnetically or inductively coupled systems operate passively in LF or HF bands. These systems behave in a similar manner to a transformer system. Figure 1 shows a reader generating a time-varying magnetic field, which induces an AC voltage at the tag. The AC voltage is then rectified to a DC voltage to energize the tag's microchip [7]. Moreover, the antenna coil in both the reader and tag are LC circuits, which has the effect of maximizing the energy transfer from the reader to the tag when tuned to the right frequency. Specifically, a higher frequency translates to a lower number of turns in the antenna coil [12][11].

Once tags are energized, reader to tags communication, and vice-versa, is achieved via Amplitude Modulation (AM). The reader modulates its magnetic field amplitude according to the digital information or baseband signal to be transmitted to a tag. A tag on the other hand, transmits its ID by turning on and off its load resistor in accordance with its ID; a phenomenon referred to as load modulation. The reader senses these amplitude variations, and demodulates the transmitted ID [7][11].

2) Electromagnetically Coupled Systems: Electromagnetically coupled systems, also called backscatter systems, operate in the UHF and microwave bands. As shown in Figure 2, the reader's dipole antenna sends out a continuous electromagnetic (EM) wave containing AC power to tags [7]. As a result, a potential difference develops at tags' dipole, thereby energizing their microchip [11]. Communication from a tag to the reader is then achieved by varying the amplitude of the EM waves reflected by the tag antenna in accordance with the digital data to be transmitted; a phenomenon called backscattering [7][8].

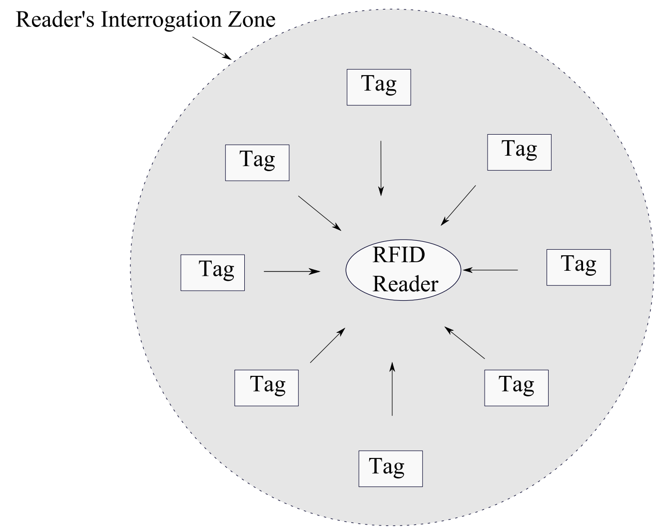


Fig. 3. The tag collision problem.

Far field backscatter systems pose many new problems which do not exist in HF or LF systems. A key problem is the reflection of the reader field due to objects with a similar dimension to the wavelength used. These reflections can cause damping or even cancelation [12][8].

B. Operating Frequency

RFID systems operate in the Industry, Scientific and Medical (ISM) frequency band that ranges from 100 KHz to 5.8 GHz. Table II [12][8][11][13][14][15] summarizes the characteristics of RFID systems based on their operating frequency.

C. Tag types

Tags are the basic building block of an RFID system. A tag consists of an electronic microchip and coupling elements. RFID tags without a microchip are called chipless tags, and promise significant cost savings since they can be printed directly on products [19][7].

There are three types of tags: passive, active and semi-passive [8][18]. Passive tags have limited computational capacity, no ability to sense the channel, detect collisions, and communicate with each other. Semi-passive tags behave in a similar manner to passive tags, but have the advantage of an on-board power source that can be used to energize their microchip. Active tags are the most expensive compared to passive and semi passive tags. Moreover, they can sense the channel and detect collisions. Table III [7][12][8][19][18] summarizes various RFID systems according to tag types.

III. Anti-Collision Protocols

Anti-collision protocols are critical to the performance of RFID systems. Figure 3 shows eight tags and a reader. Without an anti-collision protocol, the replies from these tags would collide and thereby prolong their identification. Also, collisions cause bandwidth and energy wastage.

Figure 4 classifies various anti-collision protocols in existent [7][10][20]. Broadly, they can be categorized into, space

TABLE I
NEAR VERSUS FAR FIELD COMMUNICATIONS.

Factor	Near-field	Far-field
Definition	The region between a reader antenna and one full wavelength of the magnetic field emitted by the reader's antenna [8].	The region beyond one full wavelength of the EM waves transmitted by a reader's antenna [8].
Field range	The magnetic induction range is calculated as $c/2\pi f$, where c is the speed of light and f is the operating frequency. Thus, as operating frequency increases, the magnetic field intensity decreases. The magnetic field decays as $\frac{1}{r^3}$, where r is the distance between the tag and reader measured along a line perpendicular to the reader coil's plane [11].	The range of far-field systems is constrained by the amount of energy received by a tag and the sensitivity of the reader's radio to the signal reflected by the tag. The reflected signal experiences two attenuations. The first attenuation occurs when EM waves travel from the reader to the tag and the second occurs on the waves reflected by the tag. As a result, the energy of the returning signal decays as $\frac{1}{r^4}$, where r is the distance between the reader and a tag [11].
Tag to reader communication	Amplitude modulation of magnetic field.	Amplitude modulation of reflected signals or backscattering.
Frequencies	Low Frequency (LF) , High Frequency (HF)	Greater than 100 MHz or (Ultra HF (UHF), Microwave)
Antenna	Coil	Dipole
Read range	Low $\longleftrightarrow$	High
Complexity	Low $\longleftrightarrow$	High
Data rate	Low $\longleftrightarrow$	High

division multiple access (SDMA), frequency division multiple access (FDMA), code division multiple access (CDMA), and time division multiple access (TDMA).

Briefly, SDMA protocols [7][21] spatially separate the channel using directional antennas or multiple readers to identify tags. They, however, are expensive and require intricate antenna designs. On the other hand, FDMA [7] protocols involve tags transmitting in one of several predefined frequency channels; thus, requiring a complex receiver at the reader. Lastly, systems based on CDMA [7][21] require tags to multiply their ID with a pseudo-random sequence (PN) before transmission. Unfortunately, CDMA based systems are expensive and power hungry.

TDMA protocols constitute the largest group of anti-collision protocols [7], and hence the focus on this paper. These protocols can be classified as reader driven, and tag driven. The former and latter are also called Reader-talk-first (RTF) and Tag-talk-first (TTF) respectively. Most applications use RTF protocols, which can be further classified into Aloha and tree based protocols/algorithms. Note, there is also a hybrid class, which combines Aloha and tree protocols. The basic idea behind RTF is that tags remain quiet until specifically addressed or commanded by a reader. On the other hand, TTF procedures function asynchronously. This means a TTF tag announces itself to the reader by transmitting its ID in the presence of a reader. Tags driven procedures are slow as compared to RTF procedures [20].

A. Aloha Based Protocols

We first review Aloha based tag reading protocols before discussing tree protocols in Section III-B. The following are Aloha variants in existent:

1) Pure Aloha (PA).
2) Slotted Aloha (SA).
3) Framed Slotted Aloha (FSA).
   a) Basic framed slotted Aloha (BFSA).
   b) Dynamic framed slotted Aloha (DFSA).
   c) Enhanced Dynamic framed slotted Aloha (EDFSA).

1) Pure Aloha (PA): In PA based RFID systems, a tag responds with its ID randomly after being energized by a reader. It then waits for the reader to reply with, i) a positive acknowledgment (*ACK*), indicating its ID has been received correctly, or ii) a negative acknowledgment (*NACK*), meaning a collision has occurred. If two or more tags transmit, a complete or partial collision occurs [9], which tags then resolve by backing off randomly before retransmitting their ID.

Pure Aloha based systems have several variants [22][9][23]:

- *PA with Muting*. When muting is used, the number of tags in a reader's interrogation zone is reduced after each successful tag response. Hence, muting has the effect of reducing the offered load to the reader after each successful identification. Figure 5 shows the behavior of PA with muting. Initially, tag 1 and 3's transmission collides, causing them to wait a random amount of time before retransmitting again. After identification, the reader silences read tags using the "mute" command.
- *PA with Slow Down*. Instead of being muted, a tag can be instructed using a "slow down" command to reduce its rate of transmissions, hence decreasing the probability of collision. Figure 6 shows how the reader slows tag 1 down after identification, resulting in tag 1 adapting its random back-off counter to reduce its transmission rate.
- *PA with Fast Mode*. A "silence" command is sent by the reader once it has detected the start of a tag transmission. This command has the effect of stopping other tags from transmitting. Tags are allowed to transmit again after the reader has sent an ACK command or until their waiting timer expires. Figure 7 shows PA with fast mode. Once the reader detects a transmission from tag 2, tag 1 and tag 3 are silenced and reactivated only after tag 2 has finished transmitting.
- *Other Variants*. Lastly, we can create two more variants, namely PA with fast mode and muting, and PA with fast mode and slow down by combining the respective features. These variants are shown in Figure 8 and 9 respectively. In Figure 8, tag 1 and 3 are silenced when tag 2 starts transmitting. After tag 2 is identified, it is muted. Similarly, in Figure 9, after tag 2 is identified using fast mode, it is slowed down to allow other tags to transmit.

TABLE II
CLASSIFICATION OF RFID SYSTEMS BASED ON THEIR OPERATING FREQUENCIES.

Criterion	**LF**	**HF**	**UHF**	**Microwave**
Frequency range	<135 kHz	13.56 MHz	860 - 930 MHz (1)	2.45 GHz
Physical coupling	Inductively-coupled systems.		Backscatter systems.	
Tag to reader communication	A tag uses load modulation to retrieve its ID and uses AM during transmissions.			
Tag characteristics	Passive		Active, passive, semi-passive	Active, passive
Communication boundary	Near Field		Far Field (2)	Far Field
Approximate read range (passive tags)	2m [14]	0.1m - 0.2m [15]	4m - 7m (3)(4)	1m (4)
Standards specifications	ISO 18000-2	ISO 18000-3 Auto ID HF Class 1	ISO 18000-6 Auto , Class 1 ID Class 0, Class 1	ISO 18000-4
Antenna components	Coil (> 100 turns) and capacitor.	Coil (< 10 turns) and capacitor.	Dipole antenna.	Dipole antenna.
Antenna technology	Air-core or ferrite-core coil	Perforated, printed, etched	Perforated, etched, printed	Printed antenna, etched
Effect on human body and water	None	Attenuation	Attenuation	Attenuation
Effect of metal	Disturbance	Disturbance	Attenuation	Attenuation
Data transfer rate	< 10 kbit/s	< 100 kbit/s	< 100 kbit/s	< 200 kbit/s
Cost considerations [12]	A larger antenna is required as compared to other RFID systems, resulting in high tag cost.	Less expensive than LF tags. Best suited for applications that require moderate range.	UHF tags are cheaper than LF or HF tags due to recent advances in IC design.	Microwave systems are expensive as compared to LF, HF and UHF RFID systems.
Typical RFID Applications [12]	Animal tagging, access control, vehicle identification, and container tracking in waste management.	Access control, smart, cards, item tagging, ticketing, document tracking, baggage control, laundries, and libraries.	Baggage handling, toll collection and supply chain management.	Electronic toll collection, real time goods tracking and production line tracking.
No. of tags read per second	Lowest ←→ Highest			
Tag power consumption	Lowest ←→ Highest			
Passive tag size	Largest ←→ Smallest			
Orientation sensitivity	Least ←→ Most			
Bandwidth	Lowest ←→ Highest			

(1) Japan has announced the allocation of the 950 MHz UHF frequency band [16].
(2) Recently, many UHF proponents are considering Near Field UHF band [17].
(3) Semi passive tags operate on UHF and have a range of 60-80m [18].
(4) Active tags operate on UHF or Microwave bands and have a range of more than 100m [18].

2) Slotted Aloha (SA): In Slotted Aloha (SA) based RFID systems, tags transmit their ID in synchronous time slots. If there is a collision, tags retransmit after a random delay. The collision occurs at slots boundary only, hence there are no partial collisions [24].

Slotted Aloha also has numerous variants [22] [9] [23]:

- *SA with Muting/Slow Down*. The principle operation is similar to PA with muting/slow down, but operates in a slotted manner.
- *SA with Early End*. If no transmission is detected at the beginning of a slot, the reader closes the slot early. Two commands are used: start-of-frame (SOF) and end-of-frame (EOF). The former is used to start a reading cycle, and the latter is used by the reader to close an idle slot early. Figure 10 depicts how early end is used to terminate idle slots.
- *SA with Early End and Muting*. The reader sends a mute command whenever it successfully identifies a tag; thereby, reducing the number of responding tags. On the other hand, if the reader detects no transmission after a small period of time, it closes the slot early using the EOF command.
- *SA with Slow Down and Early End*: This combines slow down with the early end feature.

In summary, there are four key features being used to increase the performance of Pure and Slotted Aloha based tag reading protocols: i) muting, ii) slow down, iii) early-end, and iv) fast mode. To recap, fast mode is only used in conjunction with Pure Aloha variants to reduce their vulnerability period. Early end is used by slotted Aloha variants to reduce idle listening where idle slots are terminated early. Lastly, muting and slow down have the effect of reducing the offered load to the reader.

3) Framed Slotted Aloha (FSA): In PA and SA based systems, a tag with a high response rate will frequently collide with potentially valid responses from other tags. Therefore, FSA protocols mandates that each tag responds only once per frame. The following sections describe various FSA variants.

Basic Frame Slotted Aloha (BFSA): BFSA has four variants. They are, 1) BFSA-non muting, 2) BFSA-muting, 3) BFSA-non-muting-early-end, and 4) BFSA-muting-early end. Note, the term "basic" refers to the frame size being fixed throughout the reading process. In BFSA-non muting, a tag is required to transmit its ID in each read round. In non-muting variants, the reading delay is dependent on the confidence level α ,where α=0.99 indicates 99% of the tags have been read successfully. The number of read cycles R needed to read a tag set with α confidence level is given by [25],

$$R \geq \left\lceil \frac{\log(1-\alpha)}{\log\left(1-\frac{Np_1}{n}\right)} \right\rceil \tag{1}$$

TABLE III
CLASSIFICATION OF RFID SYSTEMS ACCORDING TO TAG TYPES.

RFID Systems	Passive	Semi-Passive	Active
Tags	Passive tags have no power source and on-board transmitter. They use the power emitted from the reader to energize and transmit their stored data to the reader.	Semi-passive tags use an on-board power source to activate a tag's microchip. However, for data transmissions, backscattering is used.	Active RFID tags have an on-board power source such as a battery or solar power. The power source is used to to transmit data to a reader. Hence, they do not rely on the reader's emitted power for data transmissions.
Transceiver on Board	No		Yes
Communication Model	Reader talks first (RTF).		Tag talks first (TTF). The presence of a reader is not necessary for data transmissions.
Communication Principle	Either inductive coupling or backscatter (Near or far Field)	Backscatter (Far Field)	Neither backscatter nor inductive coupling. Tag generates electromagnetic waves on their own.
Tag to Reader Communications	Communication from reader to tags is achieved by modulating electromagnetic or magnetic waves.		Tags have an on-board transmitter and does not rely on a reader's waves.
Reader to Tag Communication	Communication from reader to tags is achieved by turning electromagnetic or magnetic energy waves off for short gaps of time. Tags detect these gaps as commands sent by the RFID reader.		Tags are able to communicate independently, and do not rely on the reader.
Operating Frequency	LF, HF, UHF, Microwave	UHF	UHF, Microwave
Tag size	Thin, flexible		Large, bulky
Read Range	0.1m - 7m	60m - 80m	More than 100m
Tag Cost (USD) [18]	0.15 - 1	0.75 - 2.00	10 - 100
System Cost	Lowest ⟷ Highest		
System Complexity	Lowest ⟷ Highest		

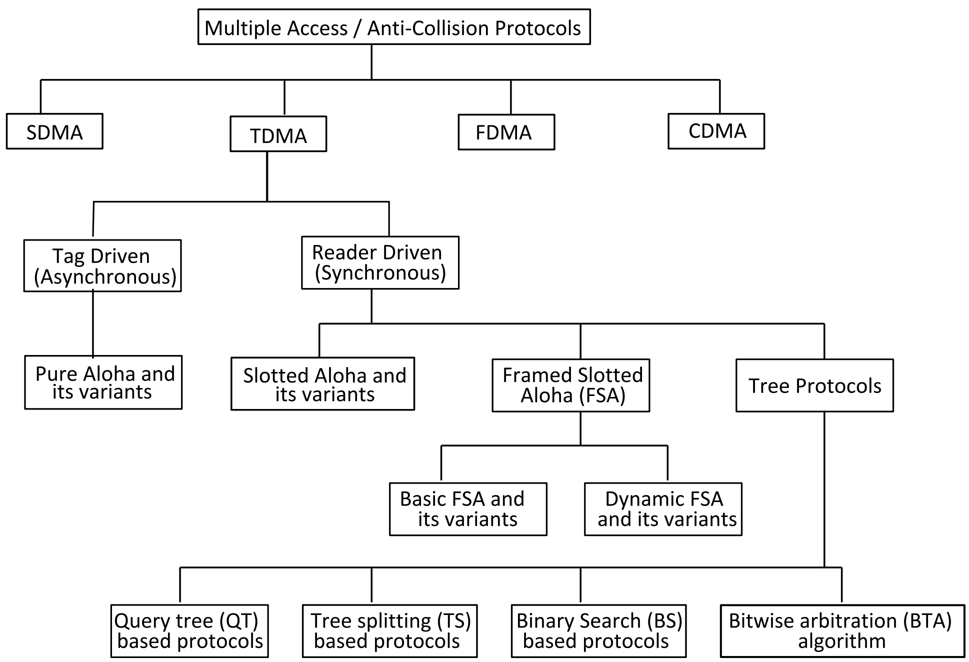


Fig. 4. Classification of tag reading or anti-collision protocols.

where N is the frame size, n is the number of tags, and the probability of having a successful transmission is $p_1 = \left(1 - \frac{1}{N}\right)^{n-1}$. To obtain an integral value, and avoid conservative delay values, Equ. 1 uses the ceil function.

For BFSA-Muting, the number of tags reduces after each read round, since tags are silenced after identification. When a read round is collision free, the reader concludes that all tags have been identified successfully.

BFSA-non-muting-early-end and BFSA-muting-early-end variants incorporate the early-end feature. Specifically, the reader closes a slot early if no response is detected at beginning of a slot.

BFSA non-muting suffers from an exponential increase in identification delay when the number of tags is higher than the frame size [26]. To address this problem, Hwang et al. [27] present a BFSA variant that limits the number of responding tags. The reader achieves this by sending a bitstring to the tags. Tags then compare a part of their ID with the said bitstring, and those with a smaller value reply. A key observation is that when the number of tags is much smaller than the frame size, restricting tag responses increases identification delays. Therefore, the authors define a threshold based on the ratio of collision slots and the frame size to decide if restricting tag responses is necessary.

A new approach, called detection and jump, is presented by Wang et al. [28]. The reader precedes each jump frame with a detection frame that has 4-bit sized slots; the detection frame is basically a reservation frame for the upcoming jump

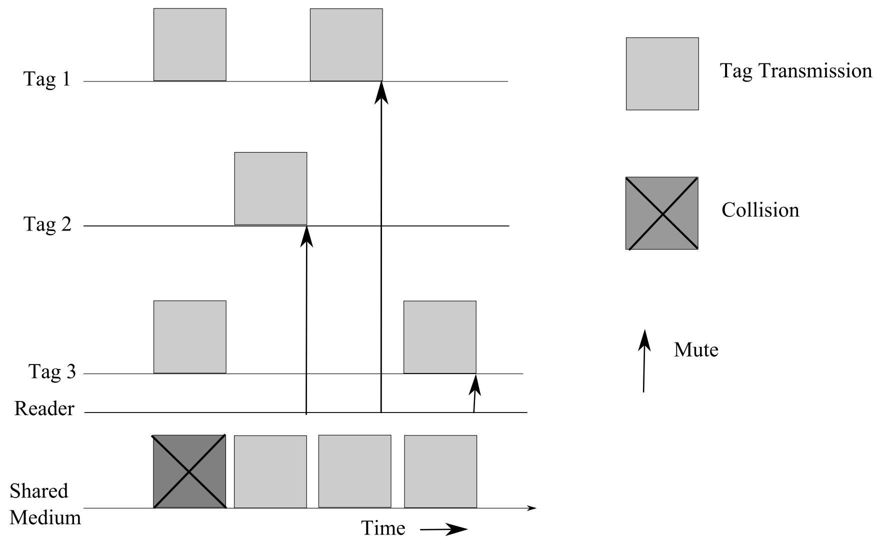


Fig. 5. Pure Aloha with muting.

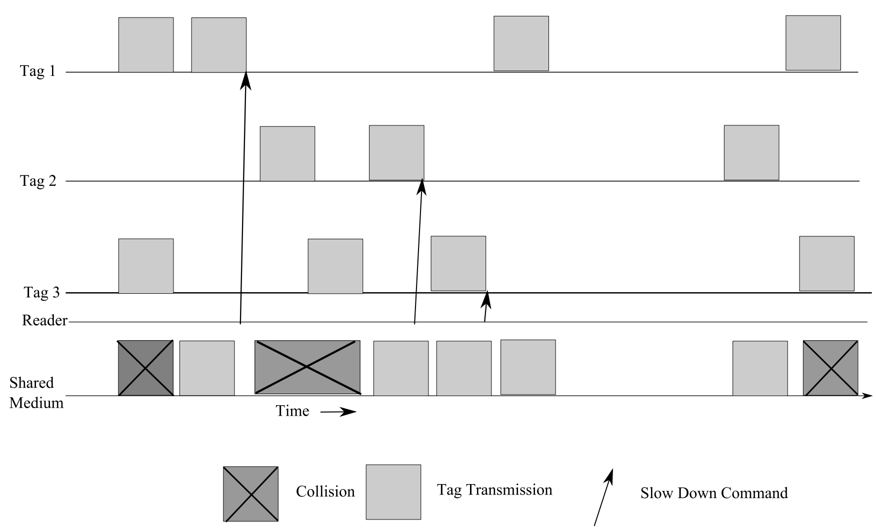


Fig. 6. Pure Aloha with slow down.

frame. Tags respond with a 4-bit random sequence in the detection frame. If they are successful, the reader informs them to transmit their ID in upcoming jump frame. Note, the size of the jump frame corresponds to the number of successful transmissions in the detection frame. Moreover, the jump frame is collision free.

Dynamic Frame Slotted Aloha (DFSA): FSA protocols with variable frame sizes are called dynamic framed slotted Aloha (DFSA) [7]. Similar to BFSA, DFSA operates in multiple rounds, and it can also incorporate the early-end feature. The key difference, however, is that in each read round, the reader uses a tag estimation function to vary its frame size [26].

A tag estimation function calculates the number of tags based on feedback from a reader's frame, which include the number of slots filled with zero (c_0), one (c_1) and multiple tag responses (c_k). This information is then used by the function to obtain a tag estimate, and hence the optimal frame size N for a given round. Here, the optimal frame size is one which promises the maximum system efficiency and minimum identification delay. Theoretically, the optimal frame size is equal to the number of tags [26].

In the following sections, we review various tag estimation functions, each of which defines a new DFSA variant.

a) Vogt [29][30]: Vogt presents two tag estimation functions, denoted as Vogt-I and Vogt-II. Vogt-I is based on the principle that during collisions, at least two tags are involved, hence the tag estimate is $c_1 + 2c_k$. On the other hand, Vogt-II is based on Chebyshev's inequality and aims to minimize the distance ε_{vd} between an actual read result vector $< c_0, c_1, c_k >$ and the theoretically computed result $< a_0^{N,n}, a_1^{N,n}, a_k^{N,n} >$; as represented by Equ. 2.

$$\varepsilon_{vd}\left(N, c_0, c_1, c_k\right) = \min_t \left| \begin{pmatrix} a_0^{N,t} \\ a_1^{N,t} \\ a_k^{N,t} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \tag{2}$$

In Equ. 2, the elements of the vector $< a_0^{N,t}, a_1^{N,t}, a_k^{N,t} >$ correspond to the expected number of empty slots, slots filled with one tag, and slots with collisions, respectively. With a

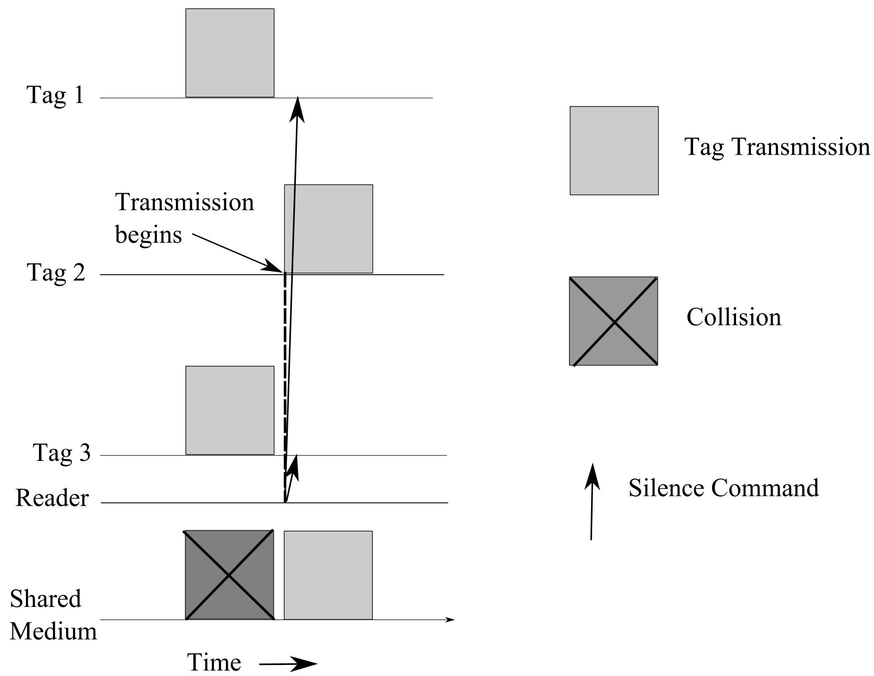


Fig. 7. Pure Aloha with fast mode.

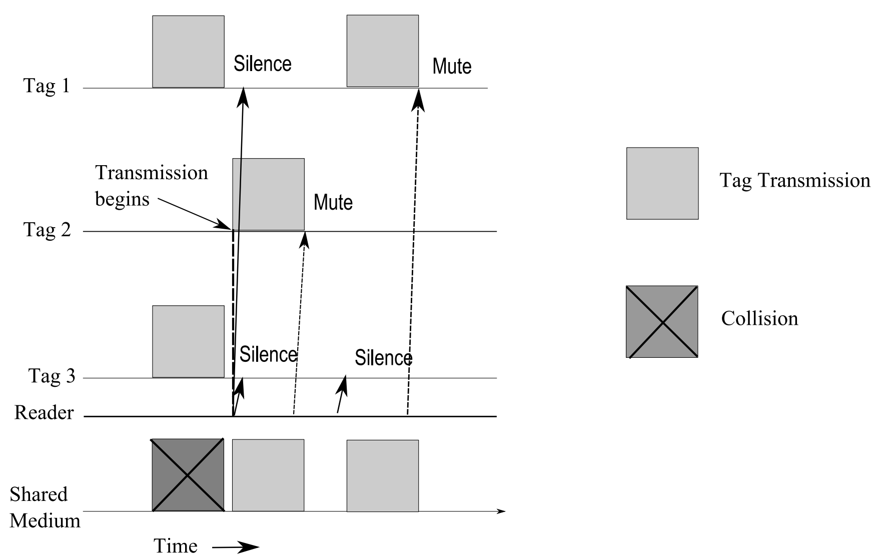


Fig. 8. Pure Aloha with fast mode and muting.

frame size of N, and the number of tags t, the expected number of slots filled with r responding tags is given by,

$$a_r^{N,t} = N \times \binom{t}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{t-r} \tag{3}$$

Vogt also proposed a set of frame sizes promising lower identification delays for a given tag range. They are shown in Table IV. For example, a frame size of sixteen is considered optimal when there are one to nine tags.

TABLE IV
OPTIMAL FRAME SIZES FOR A GIVEN TAG RANGE.

Frame Size (N)	Low (n)	High (n)
16	1	9
32	10	27
64	17	56
128	51	129
256	112	∞

b) Zhen et al. [25]: This function is based on computing the expected number of collisions in each slot, which Zhen et al., derived to be 2.39. In other words, 2.39 tags on average are involved in a collision. Thus, the number of estimated tags is $c_1 + 2.39c_k$. In addition, Zhen et al. propose to overestimate the tag set, since doing so lowers identification delays. Based on their experimentations, they proposed $1.4 \times (c_1 + 2.39c_k)$ as a tag estimate. On the other hand, for muting environments, they proposed $0.65 \times (c_1 + 2.39c_k)$.

c) Cha et al. [26]: The authors present two tag estimation functions for muting based RFID environments. Cha-I estimates tags by computing the ratio of the number of slots

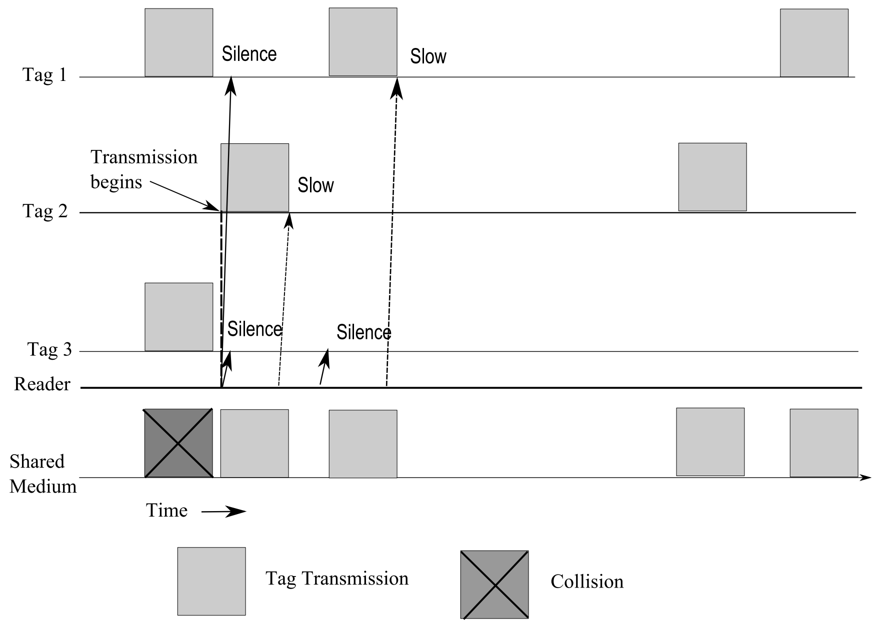


Fig. 9. Pure Aloha with fast mode and slow down.

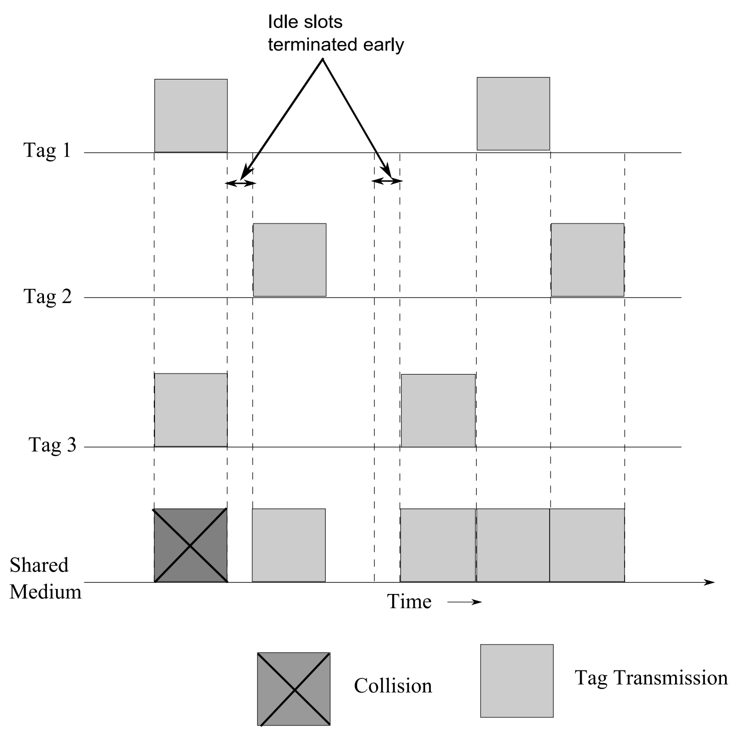


Fig. 10. Slotted Aloha with early end.

with collisions and the frame size, and is given by,

$$C_{ratio} = 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right) \quad (4)$$

where n is the tags to be estimated. C_{ratio} is computed after a read round as $C_{ratio} = \frac{c_k}{N}$. In Cha-II, a tag estimate is simply $2.39c_k$.

d) Khandelwal et al. [31]: The authors propose to estimate the number of tags using,

$$n = \frac{log\left(\frac{c_0}{N}\right)}{log\left(1 - \frac{1}{N}\right)} \quad (5)$$

Here, N is the current frame size. Note, Equ 5 cannot be applied when $c_0 = 0$. When this happens, the tag estimate is $n = c_1 + 2c_k$. Lastly, Khandelwal et al. proposed to set the frame size to $1.943 \times n$ times the estimated number of tags.

e) Floerkemeier [32][33]: Here, there are two estimation functions of interest: Floerkemeier-I and Floerkemeier-II. These functions estimate tags based on the Bayesian transmission strategy proposed by [34]. In Floerkemeier-I, a reader not only considers read results in the current read round, but also records those in the last frame to determine the frame size of the next read round. On the other hand, Floerkemeier-II updates the frame size as the frame progresses, i.e., slot-by-slot according to read results in the last and current slot. It restarts the current frame if it is non-optimal.

f) Kodialam et al. [5]: The authors proposed an estimation function that computes the expected number of idle and single response slots by inserting $r = 0$ and $r = 1$ in Equ. 3. The resulting equations are then used to derive two estimators, called zero estimator (ZE) and collision estimator (CE),

$$ZE = e^{-(n_0/N)} = \frac{c_0}{N} \tag{6}$$

$$CE = 1 - \left(1 + \frac{n_k}{N}\right) e^{-(n_k/N)} = \frac{c_k}{N} \tag{7}$$

In Equ. 6 and 7, n_0 is the tag estimate obtained from ZE, and n_k is the tag estimate computed from CE, respectively. The values of c_0 and c_1 are obtained by observing the number of idle slots and slots with single response. They are then used to solve ZE for n_0 and CE for n_k. If $n_0 < n_k$, then the tag estimate is n_0, otherwise it is n_k . The authors assume that the estimation phase is separate and precedes the identification phase. In addition, slots in the estimation phase are only 10-bits long.

g) Chen et al. [35]: The authors introduce two estimation functions, Chen-I and Chen-II. In the former, the authors compute the probability of exactly k tags in m slots as [36],

$$\begin{aligned} p(k,m) \quad &= \quad \frac{(-1)^m N!n!}{m!N^n} \times \\ \sum_{j=m}^{\min(N,(\lceil n/k \rceil))} &(-1)^j \frac{(N-j)^{n-jk}}{(j-m)!\,(N-j)!\,(n-jk)!\,(k!)^j} \end{aligned} \tag{8}$$

Using Equ. 8, the authors calculate the probability of exactly m slots with zero tag responses, i.e. $k = 0$. The actual value of m is c_0, which is obtained from the reader's feedback. The probability equation is then solved for the value of n, which is the tag estimate.

In the latter, i.e., Chen-II, the function computes the expected number of slots filled with zero and a single tag using Equ. 3. The results, denoted as E and S, are then fed into the following equation,

$$n = (N - E - 1)\frac{S}{E} \tag{9}$$

where N is the frame size. Equ. 9 is then solved for the tag estimate n.

h) Q protocol [37]: The proposed tag estimation function requires the reader to increment and decrement the frame size with a constant. A reader initially broadcasts a query command that contains a slot counter Q and a frame of size 2^Q. Q is an integer between zero and eight. Tags choose a slot randomly from 0 to $2^Q - 1$. The reader then increments or decrements Q by a constant c, where $0.1 \leq c \leq 0.5$, for each collision or idle slot respectively. Slots with a single response do not change Q. The resulting value of Q is then used to determine the frame size of the next round [28][32][33].

i) Discussions: In general, two methodologies are used for tag estimation. The first is based on computing a tag estimate using a fixed multiplier. This is called static estimation. The function Cha-I, Zhen, Q-protocol and Vogt-I belong to this methodology. On the other hand, functions which derive tag estimates using probabilistic or statistical methods are called dynamic estimation. Chen-I, Chen-II, Cha-II, Vogt-II, Khandelwal, Kodialam, FloerkemeierI, FloerkemeierII are examples of this methodology.

Among static and dynamic estimation functions, dynamic ones yield more accurate estimates as the number of tags increases. This is because dynamic estimates are obtained via statistical inferences with no reliance on a fixed multiple for collided slots. On the other hand, static estimates become erroneous as the number of tags increases beyond the given frame size. In other words, dynamic estimation functions are more accurate towards imprecise knowledge of tags for a wider tag range, whereas static estimation functions are better for lower tag ranges [29][35][30].

Estimation functions can also be classified according to their consideration for the muting feature. Among those studied, Cha-I, Cha-II, Chen-I, Chen-II, Floerkemeier-I, and Floerkemeier-II consider muting while the rest do not.

The computational requirements of tag estimation functions vary for each methodology. Static estimation techniques are simpler to implement and have low computational requirements. The computation only involves simple additions and multiplications. On the other hand, dynamic estimation techniques have higher computational requirements since they need to evaluate theoretical values and compare them to read values.

Vogt-I, Cha-II, Q-Protocol, and Zhen estimate tags using simple calculations involving additions and multiplications. Relatively higher computations are required for Chen-II, Khandelwal, Kodialam, Floerkemeier- I, and Floerkemeier-II because these functions involve the calculation of factorials and fractions. Vogt- II, Cha-I and Chen-I have the highest computational requirements since they involve recursions.

Lastly, it is important to note that very little works have conducted a comprehensive study on the accuracy of current tag estimation functions. As a result, amongst the tag estimation functions surveyed, it is unclear which is the best or most accurate. For this reason, we have chosen to evaluate tag estimation functions' accuracy in [38]. That is, the error in tag estimates. Using accuracy as a metric helps evaluates tag estimation functions by their own merit, and hence provides an unbiased indication of their performance. Specifically, in [38], we compared the accuracy of Vogt-I, Vogt-II, Cha-I, Cha-II, and Zhen, and found that Vogt-II , which is based on Chebychev's inequality, achieves the best accuracy for a wide range of tags. On the other hand, a function proposed by Cha-I is more accurate when the number of tags increases beyond the current frame size.

Enhanced Dynamic Framed Slotted Aloha (EDFSA): A limitation of DFSA variants is that the frame size is bounded to a maximum value of 256 [29] or 512 [39]. If the number of tags exceeds this value, persistent collisions become a key issue. To this end, Lee et al. [40] propose an enhanced version of DFSA, called enhanced-DFSA or EDFSA, where tags are

TABLE V
EDFSA FRAME SIZES. n DENOTES THE NUMBER OF TAGS, N IS THE FRAME SIZE, AND M IS THE NUMBER OF TAG GROUPS.

Number of tags (n)	Frame Size (N)	M
$1-11$	8	1
$12-19$	16	1
$20-40$	32	1
$41-81$	64	1
$82-176$	128	1
$177-354$	256	1
$355-707$	256	2
$708-1416$	256	4
$1417-2831$	256	8

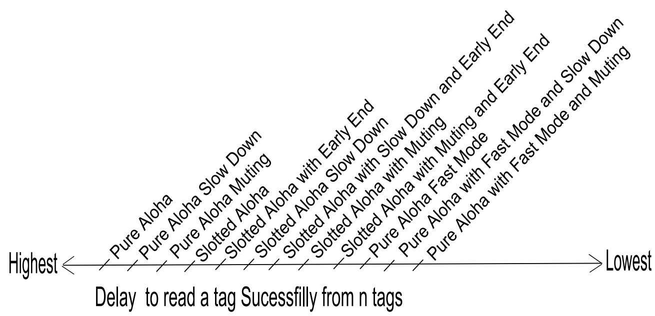


Fig. 11. Reading delay of pure/slotted Aloha variants.

divided into M groups if the tag population is larger than the maximum frame size available. Table V shows the value of M for a given tag range.

In Table V, Lee et al. also propose frame sizes for varying tag ranges to achieve maximum system efficiency. The value of M is one when the number of tags is lower than 354. However, when the number of tags increases, the modulo operation comes into effect, which divides responding tags into M groups. The reader then read tags on a group-by-group basis.

Lastly, similar to BFSA and DFSA, EDFSA can also incorporate the early end and muting feature [41].

4) Discussions: Table VI summarizes our observations pertaining to Aloha based protocols. We see that the performance of Aloha based protocols increases as we move from PA to DFSA variants. However, this performance improvement is at the expense of increased system cost and complexity.

The most suitable protocol depends largely upon the application in question. If low cost and complexity is desired, then PA variants are suitable. On the other hand, DFSA variants are ideal if high speed, accuracy, and efficiency are of concern. Overall, DFSA variants are the most popular due to their adaptability to varying loads and high system efficiency.

Lastly, in [42][41], we have conducted a comprehensive study of Aloha based protocols. Figure 11 and 12 summarize our results. From Figure 11, pure Aloha with fast mode has the lowest reading delay among pure/slotted Aloha variants. On the other hand, DFSA/EDFSA is the fastest among framed Aloha variants, especially for large tag numbers.

B. Tree Based Protocols

Tree based protocols were originally developed for multiple access arbitration in wireless systems [44]. These protocols are able to single out and read every tag, provided each tag has a

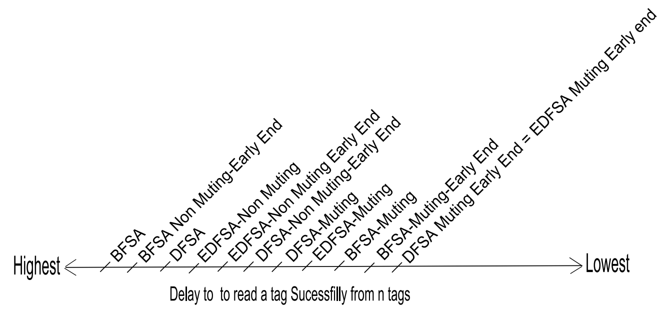


Fig. 12. Reading delay of framed Aloha variants.

unique ID. All tree based protocols require tags to have muting capability, as tags are silenced after identification. Tree based algorithms can be classified into the following categories:

1) Tree splitting (TS).
2) Query tree (QT).
3) Binary search (BS).
4) Bitwise arbitration (BTA).

1) Tree Splitting: TS protocols operate by splitting responding tags into multiple subsets using a random number generator. We present two algorithms in this category.

Basic Tree Splitting (BTS): Hush et al. [45] present BTS, an algorithm that performs collision resolution by splitting collided tags into b disjoint subsets. These subsets become increasingly smaller until they contain one tag. Identification is achieved in a sequence of timeslots. Each tag has a random binary number generator b. In addition, each tag maintains a counter to record its position in the resulting tree. Tags with a counter value of zero are considered to be in the transmit state, otherwise tags are in the wait or sleep state. After each timeslot, the reader informs tags whether the last timeslot resulted in a collision, single or no response. If there was a collision, each tag in the transmit state generates a random binary number and adds the number to its current counter value. On the other hand, tags in the wait state increment their counter by one. In the case of idle or single response, tags in the wait state decrement their counter by one. After identification, tags enter the sleep state.

As an example, let's say there are four tags: A=010, B=011, C=100, and D=110. Figure 13 depicts the identification process, and Table VII shows each tag's counter value at a given timeslot. In timeslot 1, each tag's counter is initialized to zero, meaning all tags are allowed to transmit, thus causing a collision. The reader then informs tags of the collision and tags in the transmit state split into two subsets by generating a random binary number. Tags A, B, and D have selected binary zero and therefore are allowed to transmit again, which unfortunately causes a collision in timeslot 2. At timeslot 3, only tag A has a counter value of zero, whilst the rest of the tags are in the wait state. Since tag A is the only one in the transmit state, it is identified successfully. The reader informs tags in the wait state of the single response, causing them to decrement their counter by one. In timeslot 4, tags B and D have a counter value of zero, meaning their transmission causes another collision. Tags B and D then update their counter, but experience a collision in timeslot 5. They are not identified until timeslots 6 and 7 respectively. Finally, after

TABLE VI
A COMPARISON OF ALOHA, SLOTTED ALOHA AND FRAMED ALOHA PROTOCOLS.

Criterion	Pure Aloha (PA)	Slotted Aloha (SA)	Basic Framed Slotted Aloha (BFSA)	Dynamic Framed Slotted Aloha (DFSA)
Protocol feature	A tag transmits its ID after a random time to the reader. In the event of a collision, a tag will retransmit after a random delay.	Tags transmit their ID in synchronized slots. If there is a collision, a tag responds after a random number of slots.	A tag is permitted to transmit at most once in a fixed frame (1).	A tag transmits once per frame, and the frame size varies according to tag population (2).
Tag requirements	Timer	Random number generator, timer, and synchroniza- -tion circuits.	Random number generator, and synchronization circuit. Some tags in DFSA based variants also need to generate short pseudo IDs for identification or tag estimation.	
Throughput (3)	18.4% [7]	36.8%		42.6% [43]
Disadvantages	If the offered load is increased, the number of collisions increases exponentially.	If the offered load is increased, the number of collisions increases exponentially. Also, it requires synchronization between the reader and tags.	Tags need to know the frame size in use, and they also require synchronization circuits.	Monitoring slots with single, zero or no responses, and requires a sophisticated receiver.
RTF/ TTF	TTF	RTF		
Tag cost	Least ⟷ Most			
Protocol complexity	Least ⟷ Most			
System cost	Least ⟷ Most			
(1) In Kodialam et al. [5], a tag can skip transmission in a particular frame.				
(2) EDFSA performs better than DFSA for high tag densities in non-muting environments [41].				
(3) Normalized to offered load				

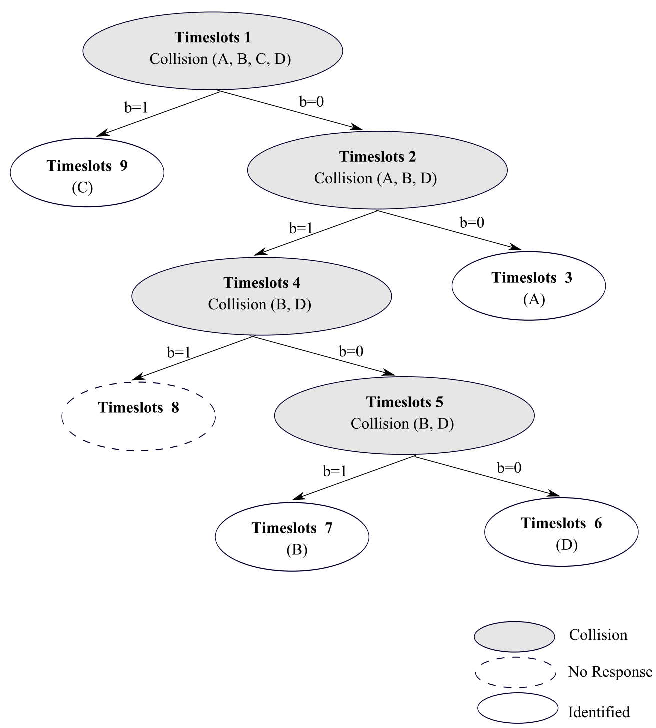


Fig. 13. Basic Tree splitting (BTS) algorithm.

an idle timeslot, tag C is identified in timeslot 9. Overall, the reader uses nine timeslots to identify all four tags.

Adaptive Binary Tree Splitting (ABTS): This algorithm, proposed by Myung et al. [46] [47],is an advancement over Hush et al. [45]'s BTS algorithm. ABTS achieves fast identification by reducing not only collisions but also unnecessary idle slots. Similar to the BTS algorithm, tags can either be in the transmit or wait state. However, unlike BTS, tags have two counters, Progressed Slot Counter (PSC) and Allocated Slot Counter (ASC). The PSC of each tag is incremented by one whenever the reader successfully identifies a tag, and ASC specifies a tag's transmitting timeslot. A tag is allowed to transmit when its ASC and PSC are equal. Moreover, identified tags have a smaller ASC compared to their PSC. As in BTS, the reader inform tags about the read result of the last timeslot. If there was a collision, tags in the transmit state or collided tags select a random binary number and add it to their current ASC. For no response or idle slots, tags in the wait state decrement their ASC by one. Lastly, if there was only a single response, tags in the wait state increment their PSC by one.

We illustrate the operation of ABTS using the tags set presented earlier. ABTS and TS share the same tree. Table VIII shows the counter value of each tag at a given tree node. Initially, the ASC and PSC value of each tag is initialized to zero. This results in a collision at timeslot 1. The tags then generate a binary random number and add the result to their ASC. In timeslot 2, tag A, B and D have equal ASC and PSC value, which causes them to enter the transmit state. As a result, their transmission collides. In timeslot 3, only tag A has equal ASC and PSC value, hence it is identified successfully. The reader then informs tags in the wait state of the successful identification in timeslot 3. Upon receiving the feedback, tags increment their PSC by one. In timeslot 4, tags B and D have equal ASC and PSC value, meaning they are allowed to transmit. Unfortunately, their transmission results in a collision. In timeslot 5, both tags B and D have a random number outcome of zero, which leaves their ASC and PSC unchanged, thus causing a collision in timeslot 5. However, in timeslot 6, only tag D has equal ASC and PSC value, which allows it to be identified successfully. Finally, tags B and C are read in timeslot 7 and 9 respectively.

Once all tags are identified, the reader ends the reading process using a terminating slot counter (TSC). The value of TSC is updated after each timeslot as follows: 1) if there was a collision, the reader increments TSC by one, 2) for an idle slot, the reader reduces TSC by one, and 3) for a slot with a single response, TSC is left unchanged. As soon as PSC becomes greater than TSC, the reader terminates the

TABLE VII
TAG'S COUNTER IN THE BTS ALGORITHM.

Time slots	Feedback	Tag Counter			
		Tag A	Tag B	Tag C	Tag D
1	Collision	0 (Transmit)	0 (Transmit)	0 (Transmit)	0 (Transmit)
2	Collision	0 (Transmit)	0 (Transmit)	1 (Wait)	0 (Transmit)
3	Identified	0 (Transmit)	1 (Wait)	2 (Wait)*	1 (Wait)
4	Collision	–	0 (Transmit)	1 (Wait)**	0 (Transmit)
5	Collision	–	0 (Transmit)	2 (Wait)	0 (Transmit)
6	Identified	–	1 (Wait)	3 (Wait)	0 (Transmit)
7	Identified	–	0 (Transmit)	2 (Wait)	–
8	Idle	–	–	1 (Wait)	–
9	Identified	–	–	0 (Transmit)***	–

$*$Tags in the wait state increment their counter by one because of collision.
$**$ Tags in the wait state decrement their counter by one because of identified tag.
$***$ Tags in the wait state decrement their counter by one because of idle response.

TABLE VIII
ADAPTIVE BINARY TREE SPLITTING (ABTS) - TSC, PSC AND ACS VALUES.

Time slot	Feedback	PSC	ASC				TSC
			Tag A	Tag B	Tag C	Tag D	
1	Collision	0	0 (Transmit)	0 (Transmit)	0 (Transmit)	0 (Transmit)	0
2	Collision	0	0 (Transmit)	0 (Transmit)	1 (Wait)	0 (Transmit)	1
3	Identified	0	0 (Transmit)	1 (Wait)	2 (Wait)*	1 (Wait)	2
4	Collision	1	–	1 (Transmit)	2 (Wait)**	1 (Transmit)	3
5	Collision	1	–	1 (Transmit)	3 (Wait)*	1 (Transmit)	3
6	Identified	1	–	2 (Wait)	4 (Wait)*	1 (Transmit)	4
7	Identified	2	–	2(Transmit)**	4 (Wait)**	–	5
8	Idle	3	–	–	4 (Wait)	–	4
9	Identified	3	–	–	3 (Transmit)***	–	3

$*$ Tags in the wait state increment their ASC by one because of collision.
$**$ ASC remains unchanged and PSC is incremented by one.
$***$ Tags in the wait state decrement their ASC by one because of no response.

reading process [46] [47]. In Table VIII, after timeslot 9, the PSC is incremented to four, which is greater than TSC, hence terminating the read process in timeslot 9.

After all tags are identified, the reader and tags preserve their TSC and ASC value. From Table VIII, the ASC value of tag A is zero, tag B is two, tag C is three, tag D is one and the reader's TSC is three. Using these TSC and ASC values, re-identification of tags can be carried out in four consecutive timeslots. This is achieved as follows. The reader first initializes PSC to zero. In the first timeslot, since the ASC for tag A is also zero, tag A enters the transmit state and is identified in the first timeslot. The PSC is then incremented by one, which equals tag D's ASC. As a result, tag D is identified in the second timeslot. Similarly, tag B and C are identified in timeslots three and four respectively.

If a new tag E is added to the tag set, it is allowed to choose an ASC value ranging from zero to TSC. If tag E selects an ASC value of two, then there will be a collision in timeslot 3. This is because both tag E and B have the same ASC value. These two tags then split into two subsets by generating a unique random binary number and are identified in either timeslots four or five depending upon their binary outcome. On the other hand, if a tag departs from the reader's interrogation zone, tags in the wait state decrement their ASC and TSC by one to eliminate idle slots.

Chen et al. [48] present a variant of the ABTS algorithm called enhanced binary tree splitting (EBTS). Their algorithm uses Manchester coding to identify the location of collided bits. If a collided bit is detected, the reader stops tags from transmitting the remaining bits of their ID. Each tag maintains a pointer that stores the location of the first collided bit. If the pointer has a value k, it means the k^{th} bit suffered a collision. In other words, all bits prior to the k^{th} bits have been received correctly. Thus, in future read requests, tags only need to transmit those bits from their ID that occur after the k^{th} bit. These bits are then identified using ABS.

2) Query Tree Algorithms: In TS variants, tags require a random number generator and a counter to track their tree position, thus making them costly and computationally complex. Query tree algorithms overcome these problems by storing tree construction information at the reader, and tags only need to have a prefix matching circuit. Numerous variants of query tree algorithms exist. They are discussed in the following sections.

Query Tree: Law et al. [49] propose query tree (QT). Each tag has a prefix matching circuit. The reader transmits a query q, and tags with a matching prefix reply to the reader. Collision occurs when multiple tags have the same prefix. In this case, the reader forms a new query by appending q with a binary 0 or 1. The reader then repeats the reading process using the augmented query.

Figure 14 shows the QT protocol being used to read the tags set presented earlier. Table IX shows the content of the reader's stack, which stores pending queries. The reader starts with a null string. Since this causes a collision, the reader pushes queries 0 and 1 onto the stack, i.e., $q = 0$ and $q = 1$. In round 2, the reader pops and transmits query 0. In our example, tag 010 and 011 have prefix 0, which causes them to transmit and collide. The reader then pushes queries 01 and 00 onto the stack. In round 3, the reader pops and transmits query 00. This query solicits no reply since there are no tags with the prefix 00. In round 4, the reader experiences a collision, since tag 010 and 011 responded to the query 01. As a result, queries 010 and 011 are pushed onto the stack. The reader

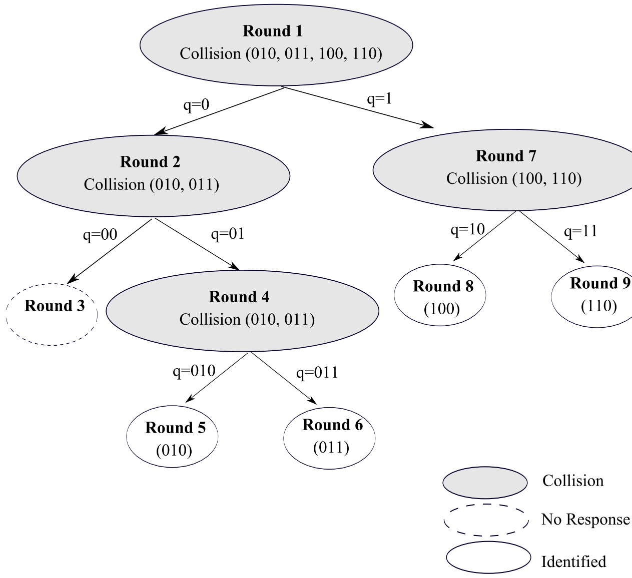


Fig. 14. The QT Algorithm.

TABLE IX
READER'S STACK CORRESPONDING TO FIGURE 11.

Round	Query q	Response	Reader's Stack
1	Empty	Collision	(0, 1)
2	0	Collision	(00, 01, 1)
3	00	Idle	(01,1)
4	01	Collision	(010, 011, 1)
5	010	Identified	(011,1)
6	011	Identified	(1)
7	1	Collision	(10,11)
8	10	Identified	(11)
9	11	Identified	Empty

then transmits query 010 in round 4, which matches tag 010. In round 5, query 011 identifies tag 011. Similarly, tag 100 and 111 are identified after the reader sent queries 10 and 11 in round 8 and 9 respectively. Overall, the reader uses nine rounds to read four tags.

Law et al. [49] also propose numerous extensions to the QT protocol. They are summarized below [49]:

Shortcutting: This extension reduces QT's identification delay by removing redundant queries. It works as follows. The reader transmits a query q, and if there was a collision, the reader appends q with 0 and 1, and pushes $q0$ and $q1$ onto the stack. The reader first transmits the query $q0$. If there was no response, the reader infers that at least two tags have the prefix $q1$. Thus, if the reader transmits $q1$, a collision will occur. Therefore, the reader removes the query $q1$ from the stack and pushes $q10$ and $q11$ onto the stack instead. Figure 15 shows the shortcutting procedure using the example shown in Figure 14. In round 2, a collision occurs for query 0. In round 3, the reader transmits query 00 but received no response. The reader then skips the transmission of query 01, and pushes queries 010 and 011 onto the stack. Tags 010 and 011 are then identified in round 4 and 5 respectively. Notice that in Figure 14 there is a collision in round 4, which does not exist when using the shortcutting extension.

Aggressive enhancement. In this extension, queries are appended with multiple bits, instead of a single bit. For example, if query q causes a collision, the reader proceeds with queries

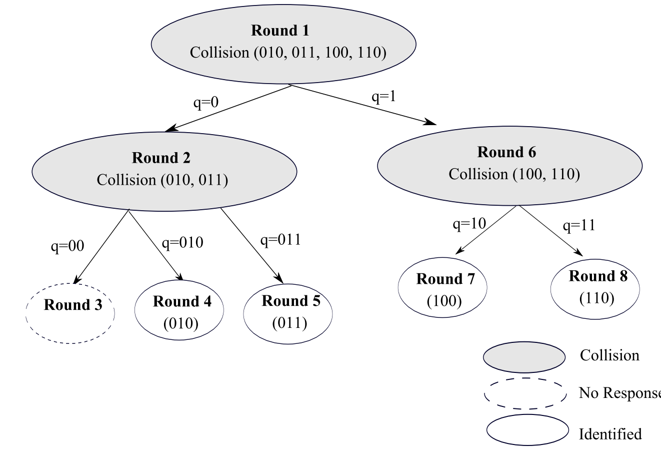


Fig. 15. QT with Shortcutting.

$q00$, $q01$, $q10$ and $q11$ directly. This approach requires more queries compared to the original QT protocol [49].

Categorization. In this QT enhancement, the reader has prior knowledge of tag IDs, thereby allowing the reader to group tags according to predefined prefixes.

QT-sl (Query-tree short-long) protocol. Here, the reader separates tag responses into short and long queries. Short queries solicit a 1-bit response from tags, while long queries cause tags to send all bits of their ID. Long queries are sent when the reader knows there is only going to be one matching tag [49].

QT-im (Query-tree incremental-matching) protocol. This algorithm reduces the number of query bits transmitted by requiring tags to remember the last query sent by the reader. For example, if the query transmitted by the reader in the last read round is q, then in the next read round, instead of sending query $q0$ or $q1$, the reader transmits 0 or 1 [49].

Lastly, Choi et al. [50] proposed a scanning based pre-processing (SBPP) technique that uses Manchester coding to locate collided bits in tag responses. The reader notifies tags the whereabouts of these collided bits, and uses a QT algorithm to identify them.

Adaptive QT (AQT): In [51][52], Myung et al. proposed a protocol, called the adaptive query tree (AQT), where the reader is required to maintain a queue Q that operates similarly to the stack in the QT algorithm. In addition, the reader is required to maintain a candidate queue (CQ) for storing queries sent in past identification rounds.

Using AQT, the earlier tags set can be identified as follows. Initially, with no past information, the tree construction of AQT is is similar to the QT protocol; see Figure 14. Once the tree is formed, leaf node 00, 010, 011, 10 and 11 are stored in CQ. The leaf nodes comprise of no response queries and those with a single tag response. To re-identify the same set of tags again, the reader uses the queries stored in CQ; i.e., 010, 011, 10 and 11.

To identify new tags, the reader relies on CQ. Consider two new tags, 111 and 000. The reader begins with query 00, which matches tag 000. Tag 010, 011, and 100 are identified using queries 010, 011 and 10 from the CQ respectively. Query 11 results in a collision between tags 110 and 111. Thus, the reader pushes queries 110 and 111 onto the stack. These two

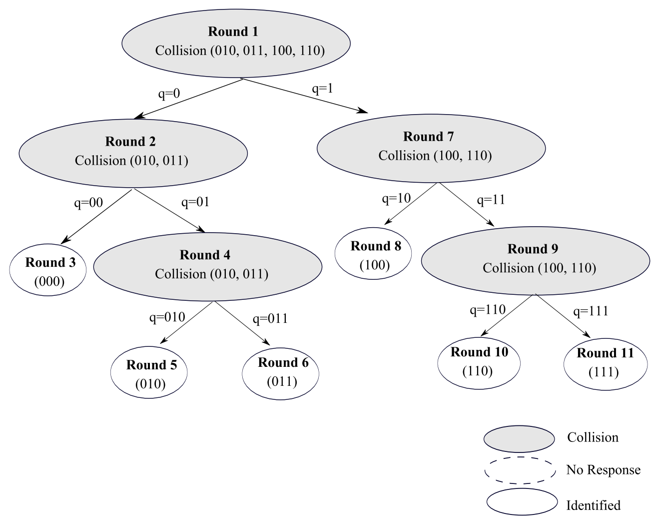


Fig. 16. AQT- new tags 111 and 000.

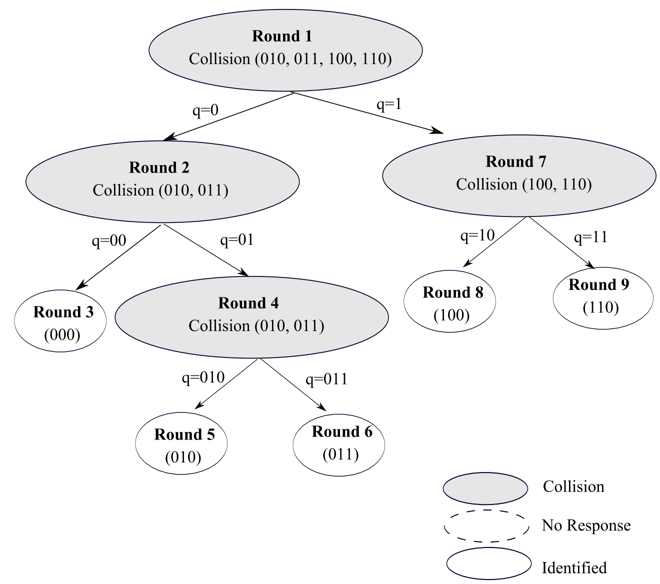


Fig. 17. AQT-departed tag 111.

queries are then used to identify tags 110 and 111. Figure 16 shows the updated tree with tags 000 and 111. Lastly, CQ is updated to store the new leaf nodes.

On the other hand, if a tag, say 111, departs, there will be no response for the query 111. This means for query 11, only tag 110 replies. The reader then replaces queries 111 and 110 with the query 11. The resulting tree only have a single response node for query 11, as shown in Figure 17.

Improved QT: Zhou et al. [53] improved the QT algorithm, referred to as IQT, by reducing the number of bits transmitted from tags to the reader when a collision occurs. The key feature of IQT is that the reader monitors tag responses in a bit by bit manner. If a collision occurs at a particular bit, the reader signals tags to stop transmitting.

QT based reservation (QTR): Choi et al. [54] proposed a QTR algorithm. The key difference to QT protocol is that tags use a 16-bit random number during the identification process. After this number is identified, the reader requests tags to respond with their complete ID.

Randomized Hashing Query Tree (RH-QT): Bonuccelli et al. [55] introduced a randomized hashing based QT approach. Each tag generates a random number from a predefined hash function using parameters sent by the reader. The reader has prior knowledge of all possible random numbers that can be generated from the hash function. The reader then uses these numbers to query tags. A tag replies if it finds that the number sent by the reader matches its own number. If multiple tags have the same random number, collisions occur. Hence, these tags will have to select a new random number, and the reader then repeats the process to identify the collided tags.

Intelligent Query Tree (IQT): This algorithm [56] exploits tags' prefix patterns, e.g., common vendor or product ID. This means a reader using IQT will first identify common prefix bits, and skips these bits in subsequent read rounds.

3) Binary Search (BS): BS algorithm [7] involves the reader transmitting a serial number to tags, which they then compare against their ID. Those tags with ID equal to or lower than the serial number respond. The reader then monitors tags reply bit by bit using Manchester coding, and once a collision occurs, the reader splits tags into subsets based on collided bits.

Figure 18 depicts a reader using BS to read the tags set presented earlier. Initially, the reader starts reading with the maximum possible tag ID value, i.e., 111. Tags with an ID value less than 111 respond, resulting in the reply XXX. This indicates all three bits have experienced a collision. The reader then transmits another query by replacing the most significant collided bit with 0, and sets the other bits to 1, i.e., the new query becomes 011. This subsequent query solicits the response 01X. The reader then sends the query 011. Only tag 010 have ID lower than 011 and therefore it is identified successfully. After that, the reader restarts the reading with query 111.

Yu et al. [57] presented a variant of BS called enhanced-BS algorithm (EBSA). The key difference to BS is that EBSA does not restart the reading process after a tag is identified. Moreover, during initialization, the reader transmits a '1' instead of sending a serial number consisting of all ones. Liu et al. [58] improved EBSA further by identifying two tags simultaneously when there is only a single collided bit.

Another enhancement to the BS protocol is called the dynamic BS algorithm (DBSA) [7]. In DBSA, the reader and tags do not use the entire length of serial number and tags ID during the identification process. For example, if a reader receives the response 01X, tags only need to transmit the remaining part of their ID since the reader has identified the prefix 01. This enhancement effectively halves the amount of data sent by the reader to tags.

4) Bitwise Arbitration (BTA) Algorithms: Researchers have proposed various BTA algorithms. Unlike TS, QT, and IDS protocols, BTA algorithms operate by requesting tags to respond bit by bit from the most significant bit (MSB) to the least significant bit (LSB) of their ID. The key feature of BTA algorithms is that bit replies are synchronized, meaning multiple tags responses of the same bit value result in no collision. A collision is observed only if two tags respond

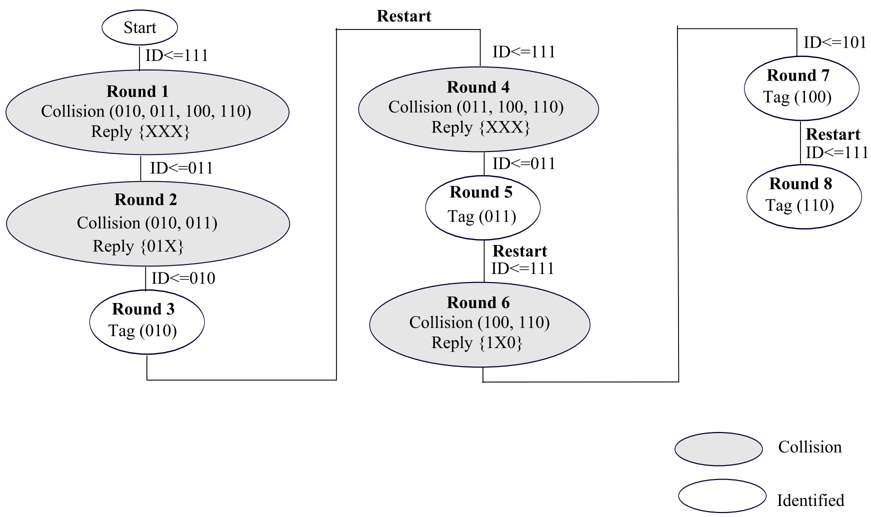


Fig. 18. The BS Algorithm.

with different bit values. Moreover, the reader has to specify the bit position it wants to read.

ID-Binary Tree Stack: The ID binary tree stack (ID BTS) [59] works by constructing a binary tree that has a height k, corresponding to the maximum tag ID with length k. Every branch corresponds to the bit of the tag ID. For any node x in the ID-binary tree, the left and right branch is labeled with binary zero and one respectively. A path from the root to an internal node represents a tag prefix, and a path from the root to a leaf node leaf node defines a unique tag ID.

The reader uses a stack to store tags' position on the tree, while a tag has a counter to record the depth of the reader's stack. Based on this counter value, a tag determines whether it is in the transmit or wait state. In other words, a counter value of zero moves a tag into the transmit state. Otherwise, the tag enters the wait state. Once a tag is identified, it enters the sleep state.

Figure 19 shows the construction of an ID binary tree for the example tag set A=010, B=011, C=100, and D=110. Table 10 shows the reader stack and tags counter. In round 1, the reader commands tags to respond with their first or MSB, which results in a collision. The reader then transmits a control bit to silence tags that responded with a binary one. After that, the reader pushes a binary one into the stack, and silenced tags increment their counter by one to record their stack position. The reader then proceeds to read tags that responded with a binary zero in round 2. The reader requests the second MSB from tag A and B, which is received correctly as both tags transmitted a bit value of one. In round 3, tags respond with their third ID bit, causing a collision. Since the tag ID in this example is three bits in length, a collision in the third bit indicates two responding tags have a third bit value of zero and one. The reader thus appends zero and one to the first two received bits, thereby identifying tag A and B successfully. After that, the reader pops binary one from the stack, which is the first bit of the silenced tags or in other words, the tree position of the silenced tag C and D. Also, tag C and D decrement their counter by one. In round 4, the reader requests the second bit from tags C and D, which ends in a collision. Similarly, the reader pushes binary one onto the stack. In round 5 and 6, the second and third ID bits of tag C are identified respectively. Finally, tag D is identified in round 8.

Bit-by-bit (BBT): Jacomet et al. [60] presented a BBT arbitration method where a separate channel is used for binary zero and one. When requested, each tag transmits the specified bit in one of these channels. If the reader receives a different response from both channels, it sends a control bit silencing the subset of tags that replied with 0 (or 1). On the other hand, if the reader receives a response in only one of the two channels, a bit is identified successfully. Similar to ID-BTS, the reader has a stack and each tag has a counter to store its tree position.

Modified bit by bit binary tree (MBBT): This algorithm, proposed by Choi et al. [61], operates in a similar manner to the BBT algorithm. The key difference is that MBBT does not use multiple timeslots to receive binary 0s and 1s.

Enhanced bit by bit binary tree (EBBT): Choi et al. [61] also proposed the EBBT algorithm. In EBBT, a reader first requests tags to respond with their complete ID. The assumption here is that tags responses are synchronized. From these responses, the reader identifies collided and collision-free ID bits. For example, let's say there are three tags: 010, 100, and 110. Initially, the reader requests tags to respond with their entire ID, which resulted in the response XX0, indicating the first two bits have experienced a collision. The reader then uses MBBT to identify the collided bits.

Bit query (BQ): Kim et al. [62] [63] propose a bit query (BQ) algorithm. A reader transmits a bit query q to tags. Tags with their prefix matching the query q respond with the bit that is adjacent to the requested prefix. Other tags deactivate themselves. If the reader receives a tag's bit response successfully, that bit is sent as the next query. However, if there is a collision, the reader uses bit zero as the next query.

Let's demonstrate the operation of BQ using the tags set earlier. Similar to the QT protocol, the reader maintains a stack and each tag has a counter. Figure 20 demonstrates the identification process for BQ. Initially, the reader transmits a bit query $q = 0$, and stores $q = 1$. This query solicits

TABLE X
READER STACK AND TAGS COUNTER.

Round	Response	Tag Counter				Reader's Stack
		Tag A	Tag B	Tag C	Tag D	
1	X	0 (Transmit)	0 (Transmit)	0 (Transmit)	0 (Transmit)	Empty
2	1	0 (Transmit)	0 (Transmit)	1 (Wait)	1 (Wait)	(1)
3	X	0 (Identified)	0 (Identified)	1 (Wait)	1 (Wait)	(1)
4	X	–	–	0 (Transmit)	0 (Transmit)	Empty
5	0	–	–	0 (Transmit)	1(Wait)	(11)
6	0	–	–	0 (Identified)	1(Wait)	(11)
7	1	–	–	–	0 (Transmit)	Empty
8	0	–	–	–	0 (Identified)	Empty

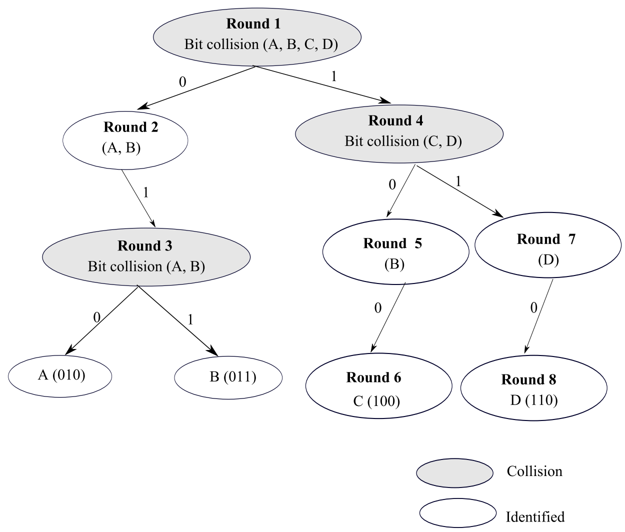


Fig. 19. ID-BTS.

a binary 1 response from tag 010 and 011, which is the bit consecutive to the requested prefix. Since there is no collision, the reader uses $q = 1$ as next query in round 2. This results in a collision due to differing bit responses received by the reader. The reader then uses $q = 0$ as the next query, and stores $q = 1$. In round 3, tag 010 is the only tag with its last bit matching the requested bit, and therefore it is identified. The reader then retrieves query $q = 1$ in round 4, which identifies the last bit of tag 011. After round 4, the reader transmits $q = 1$ and tags 100 and 110 respond with 0 and 1, thereby resulting in a collision. Similarly, due to collision, the reader transmits $q = 0$ in round 5 and stores $q = 1$. Tag 100 is identified in round 6. After that, the reader transmits the last prefix query $q = 1$, which identifies the last bit of tag 110 in round 7.

5) Discussions: Table XI compares tree protocols. Those using BTA require tags to respond bit by bit, hence are the most complex in terms of reader and tag hardware requirements when compared to QT, TS and IDS protocols. Among all tree protocols, QT protocols promise the simplest tag design.

Tree algorithms provide a deterministic approach to identify tags. On the other hand, Aloha based approaches are probabilistic in nature, simple, and promise dynamic adaptability to varying loads; unlike tree protocols which must restart their reading process if a new tag enters a reader's interrogation zone while tags are being read. Table XII shows a comparison between Aloha and tree based algorithms.

C. Hybrid protocols

Hybrid protocols are a new branch of tag reading protocols that combine the advantages of tree and Aloha protocols. A number of protocols have been proposed under this category.

1) Tree Slotted Aloha (TSA): TSA [66], an enhanced FSA protocol, uses a tree structure during the identification process. The root node of the tree denotes a frame to be transmitted in the first read round. Each tag remembers the slot number they used to transmit. At the end of a read round, if there were collisions, the reader starts a new reading cycle for each collided slot. This corresponds to adding new nodes to the tree. Each tag has a counter to remember its position in the tree. Each time a collision occurs, a new node is inserted onto the tree, and another reading cycle is initiated. The whole process is repeated until a cycle is collision free.

2) Hybrid Query Tree (HQT) Protocol: Ryu et al. [67] combined the QT protocol with a slotted random back-off mechanism. The identification proceeds as follows. A reader transmits a two bits query to tags, and tags with a matching prefix respond after a back-off delay. The duration of the back-off timer is determined as follows. Let's say there are three tags: 0100, 0101, and 0110. If the reader sends query 01, then the two bits following the prefix queried for each tag are 00, 01, and 10. These tags then set their backoff timer to zero, one and two slots respectively. Ryu et al. also proposed an enhanced HQT protocol, which uses the slotted back-off with the AQT protocol [51][52].

3) HQT variants: Shin et al. [68] proposed two algorithms that use a combination of QT and Framed Aloha protocols: Framed Query Tree algorithm and Query Tree ALOHA algorithm. In the former, the readers transmit a frame to tags, and tags choose a slot randomly. Within each slot, QT is used to identify tags. On the other hand, in the latter algorithm, the reader transmits a prefix and frame size, and tags with a matching prefix choose a slot randomly in the frame. In other words, tags with a matching prefix are identified using framed Aloha protocol.

4) Hybrid Randomized Protocol: Namboodiri et al. [69] introduce three anti-collision protocols that combined the QT protocol with DFSA. The first of these, called Multi Slotted (MS) scheme, relies on using multiple slots per query to reduce the chances of collisions. The second, called MS with Selective Sleep (MSS), uses the muting feature to silence identified tags. The third scheme, called the MS with Assigned Slots (MAS) scheme, assigns tags a specific slot in a query frame. All three protocols are capable of adjusting their frame size after each query.

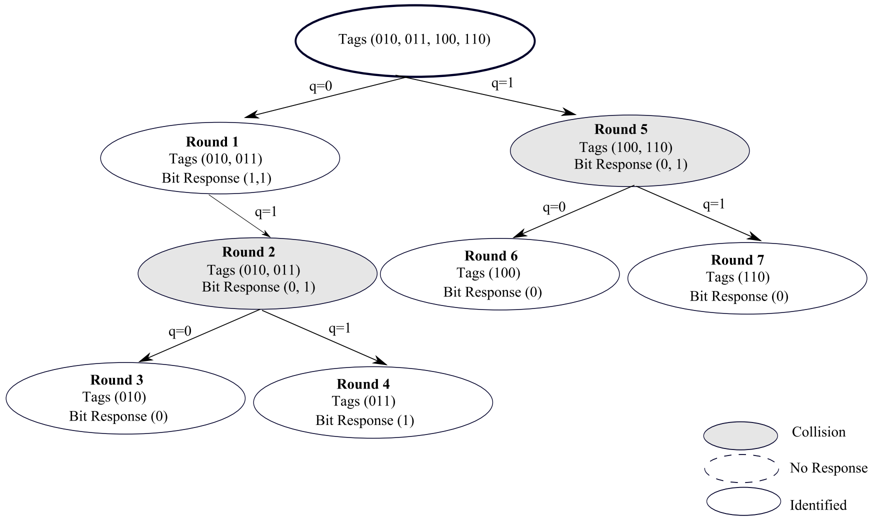


Fig. 20. Bit Query (BQ).

TABLE XI
A COMPARISON OF TREE PROTOCOLS.

Criterion	Query Tree (QT)	Tree Splitting (TS)	Binary Search (BS)	Bitwise Arbitration (BTA)
Protocol feature	The reader transmits a query, and tags with prefix matching the query respond.	The algorithm performs collision resolution by splitting collided tags into b disjoint subsets.	The reader sends a serial number to tags, and those with values less than or equal to the serial number reply.	Each tag responds in a bit by bit manner.
Tag requirements	Prefix matching and synchronization circ- -uitry.	Random number generator synchronization circ- -uits, and counters to store state information.	Manchester coding scheme, synchronization circuits.	Synchronization circuits, ability to respond in a bit-by-bit manner.
RTF/ TTF	RTF			
Time complexity (1)(3)	$O(n)$	$O(n)$	$O(logn)$	$O(2^k)$)
Message complexity (2)(3)	$2.21klogn + 4.19k$	$nlogn$	Not specified	$O(n(k+1))$
Tag cost	Least ⟷ Most			
System cost	Least ⟷ Most			

(1) Time required to identify all tags.
(2) Number of messages tags need to transmit before they are identified successfully.
(3) n denotes the number of tags and k denotes the length of tag's ID.

5) Hash-Tree Protocol: Zhang et al. [70] presented an advanced FSA algorithm that uses a hash function for slot selection in a reader's frame. The function is given in Equ. 10.

$$Hash(ID) = \frac{ID}{w}\%N \tag{10}$$

where ID is the identification code of the tag, w is a positive integer provided by the reader, and N is the frame size. The reader starts reading with a frame size L. The maximum possible frame size is L_{max}. The reader then estimates the number of tags as $2.39c_k$, where c_k is the number of collided slots. If the number of estimated tags is less than the current frame size, the collided tags are identified in sub-frames using an approach similar to MS algorithm. Otherwise, the reader expands the current frame size to 2.39 times the original frame size

6) Discussions: From the aforementioned works, it is clear that most hybrid protocols combine the QT protocol with a Aloha variant. This is because QT helps a reader separates tags into smaller groups, thereby reducing contention. Each group can then be read using a tree or an Aloha variant.

The results in [69] show that hybrid randomized protocols consume lower energy than the QT protocol. Specifically, MSS saves more energy than MS. Overall, MAS achieves the highest energy savings because it experiences fewer collisions. On the other hand, HQT [67] and its variants [68] outperform the QT protocol in terms of identification delays. Similarly, the number of collisions is lower in HQT and its variants as compared to the QT protocol. In [66], the authors show that TSA achieves a higher system efficiency compared to DFSA, EDFSA, QT, and QT with an aggressive enhancement when the number of tags is more than 60. On the other hand, when the number of tags is below 50, QT with the aggressive enhancement has the highest system efficiency.

The above results validate the advantages of hybrid protocols. Moreover, given the emergence of novel tree and Aloha variants as well as tag estimation functions, we expect researchers to propose better hybrid protocols in the near future.

IV. RFID ANTI-COLLISION STANDARDS

Two bodies are responsible for RFID air interface standards: EPCglobal [71] and international organization for standardization (ISO) [72]. EPCglobal develops industry-driven standards for international supply chain networks. ISO, on the other hand, specifies air interface specifications for tracking cattle, payment systems, contact less smart cards, and vicinity cards.

TABLE XII
COMPARISONS BETWEEN TREE AND ALOHA BASED ALGORITHMS.

Criterion	Tree protocols	Aloha protocols
Protocol feature	Tree protocol operates by grouping responding tags into subsets and then identifying tags in each subset sequentially.	Aloha based protocols require tags to respond randomly in an asynchronous manner or in synchronized slots or frames.
Newly arriving tags [64] [65]	New tags cause the re-construction of an existing tree (2).	New tags participate in collision resolution upon arrival.
Departing tags	A departing tag causes tree reconstruction (2).	Departing tags do not affect reading.
Usage	Tree protocols are mainly used in UHF and microwave RFID systems.	Aloha protocols are mainly used in LF and HF RFID systems.
Number of reader to tag commands	High	Low
Tag starvation	No	Yes (1)
Delay versus tag density [35]	Low identification delays in high tag density environments.	Low identification delays achievable only when tag density is low.
Method	Deterministic	Probabilistic
Optimum channel utilization	43% [43]	18.4% (Pure Aloha), 36.8% [7] (BFSA) [7], 42.6% (DFSA) [43]
(1) Tag starvation is largely mitigated by features such as muting and fast mode. (2) Recently proposed tree protocols such as ABS and AQT do not require tree re-construction for new or departed tags.		

Table XIII summarizes ISO standards. ISO 18000-3 “MODE 1” has two extensions. The first uses Pure Aloha, and the other relies on DFSA. ISO 18000-3 “MODE 2”, on the other hand, uses a combination of frequency and time division multiple access. ISO 14443-3 Type-A and Type B use Dynamic BS algorithm and DFSA protocol respectively. ISO-18000-6A uses Framed slotted Aloha with muting and early-end, whereas ISO-18000-6B uses ID-BTS.

Table XIV presents the standards proposed by EPCglobal. Class 0 and 1, which are developed for UHF RFID systems, use a variant of ID-BTS. Specifically, Class 0 relies on ID-BTS, whereas Class 1 uses an advanced ID-BTS, where a tag transmits eight consecutive bits to the reader, which are then identified by the reader sequentially. The class 1 HF standard, on the other hand, uses BFSA with early-end. In addition, the protocol uses partial IDs during contention. The second generation of Class 1 uses the Q protocol [37].

Lastly, Table XV shows propriety RFID specifications from Philips. I-Code and U- Code are developed for HF and UHF RFID systems respectively. I-Code uses DFSA for collision resolution and U- Code relies on the Q protocol [37]. Philips also proposed another HF standard, called *Mifare*, that uses the Dynamic BS algorithm.

From Table XIII, XIV and XV, we can see that most HF RFID standards use an Aloha variant, whereas RFID standards for UHF use both Aloha and tree protocols. In general, standards with a low bandwidth air interface rely on Aloha variant. Otherwise, systems have the flexibility to choose either tree or Aloha variants.

V. RESEARCH DIRECTIONS

From our discussions above, it is clear that researchers have studied both Aloha and tree protocols extensively. Research on Aloha protocols is shifting towards DFSA variants, specifically those that rely on a tag estimation function. From our survey, we find that dynamic estimation schemes to be the most promising because of their higher accuracy for a given tag range. However, further research is required to reduce their considerable computational cost and memory requirements.

For tree protocols, QT variants have had a number of advances. This is mainly due to their simpler tag designs that only require a prefix matching and a synchronization circuit. A key disadvantage of QT protocols, however, is that the length of a query is proportional to the depth of the constructed tree. Another problem is that identification delay increases with ID size. This issue becomes critical when the EPC adopts 256 bit IDs [37]. The current approach to address long IDs is by using randomly generated pseudo IDs [55] [54]. The advantage of such an approach is that it involves minimal data exchange between the reader and tags, and uses shorter IDs, which reduces tree depth. From our survey, an interesting observation is that, except for [55][54][78], existing tag reading protocols do not yet incorporate pseudo IDs. Therefore, an interesting research direction is to analyze the performance gains to be had if protocols use pseudo IDs.

Hyrbid protocols, i.e., those that combine Aloha and tree protocols, are becoming popular [66][67][69][70]. To date only a handful of hybrid protocols exist. Moreover, given the number of Aloha and tree protocol variants, a challenging research problem is determining the combinations that have the highest reading rate.

Apart from the aforementioned issues, a recent development in RFID systems is their integration with sensor nodes to create RFID-enhanced wireless sensor networks (WSNs) that can be deployed randomly to identify RFID tagged objects [79][80][81][82]. A key problem in such networks is the energy constraint imposed by sensor nodes. To put this in perspective, in [42], we have analyzed the energy consumption of a sensor mote with an RFID reader. We observed that an RFID reader while scanning 96-bits of tag ID consumes higher energy compared to a sensor node receiving and transmitting the same number of bits. Moreover, as a reader’s scanning/reading duration increases, so does its energy consumption.

To this end, we have conducted an energy efficiency analysis of Aloha protocols in [42] and [41]. Our results are summarized in Figure 21 and 22. We can see that Pure Aloha with fast mode and muting consumes the lowest energy among all pure Aloha variants. On the other hand, for Framed Aloha protocols, DFSA and EDFSA with muting and early end

TABLE XIII
ISO RFID STANDARDS [73][9][74][72][75].

Standard	Frequency	Protocol Used
ISO 18000-3 “MODE 1”	HF	There are two extensions: Pure Aloha and Dynamic Framed Slotted Aloha.
ISO 18000-3 “MODE 2”	HF	This protocol is a combination of both frequency and time division multiple access (FTDMA). A tag has a choice of eight reply channels. After selecting a channel, the node uses slotted Aloha to access the channel. An extension here is to combine slotted Aloha with muting and slow down.
ISO 14443-3 Type-A	HF	Dynamic BS Algorithm (DBSA).
ISO 14443-3 Type-B	HF	Dynamic Framed Slotted Aloha.
ISO-18000-6A	UHF	Framed slotted Aloha with muting and early-end.
ISO-18000-6B	UHF	ID-BTS.

TABLE XIV
EPCGLOBAL RFID STANDARDS [37][39][71] .

Standard	Frequency	Protocol Used
EPCglobal Class 0	UHF	ID-Binary Tree Stack (ID-BTS).
EPCglobal Class 1	UHF	Advanced ID-BTS. The protocol is an advancement of ID-BTS, where a tag responds to the reader with eight consecutive bits, which the reader then reads sequentially.
EPCglobal Class 1 Gen 2	UHF	Q Protocol.
EPCglobal Class 1	HF	Basic Framed Slotted Aloha with early-end. In addition, the protocol uses partial IDs during identification.

TABLE XV
PROPRIETARY PROTOCOLS [15][76][77].

Standard	Frequency	Protocol Used
Philips I Code	HF	Dynamic Framed Slotted Aloha.
Philips U Code	UHF	Q Protocol.
Philips Mifare	HF	Dynaic BS Algorithm (DBSA).

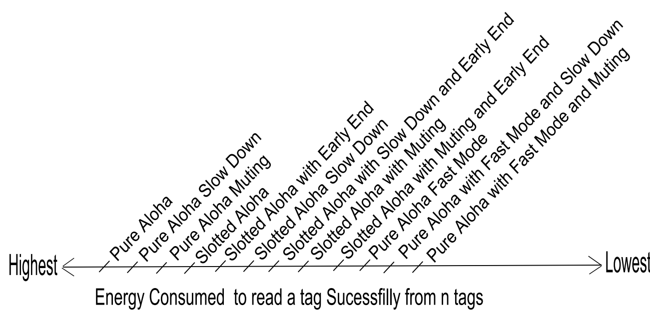


Fig. 21. Energy consumption analysis of pure/slotted Aloha variants.

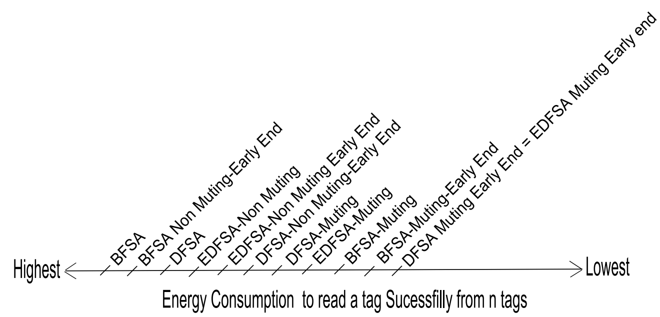


Fig. 22. Energy consumption analysis of Framed Aloha variants.

consume the lowest energy, specifically when number of tags is high.

A key parameter that impacts the energy consumption of Aloha protocols is collision. As shown in Figure 23 and 24, both pure Aloha with fast mode and DFSA/EDFSA with muting and early end has the lowest energy consumption due to collisions, especially for large tag numbers. Note, DFSA with muting and DFSA with muting and early end have similar energy consumption in collisions. This is because, the early end feature only impacts the energy consumption in idle listening and does not reduce the number of collisions [42][41].

To analyze the impact of Aloha protocols on the battery lifetime of a node, we derived the battery lifetime of Aloha protocols, and they are shown in Figure 25 and 26. Among pure/slotted Aloha variants, pure Aloha with fast mode has the highest lifetime, whereas DFSA/EDFSA with muting and early end has the highest lifetime among framed Aloha variants. Although our results in [78] show that Aloha based protocols are suitable for RFID enhanced WSNs, the viability of tree protocols in WSNs remain an open issue.

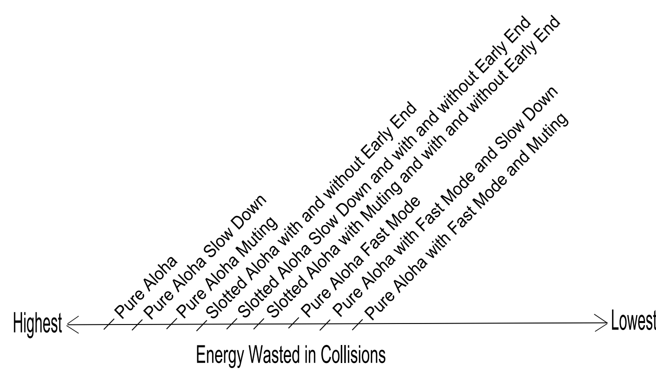


Fig. 23. Energy wasted in collisions by pure/slotted Aloha variants.

Another important consideration in RFID-enhanced WSNs is the ability to track tags. This problem is particularly acute when tag population changes frequently. Unfortunately, existing protocols are inefficient and not scalable. They ei-

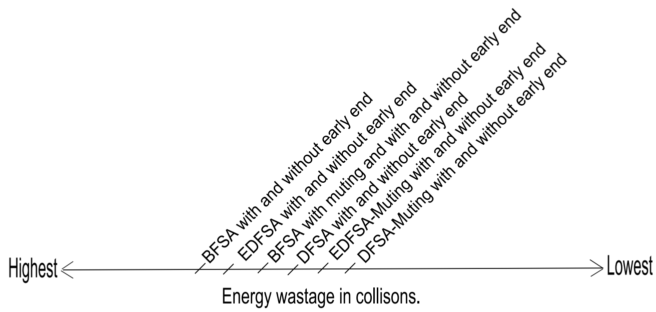


Fig. 24. Energy wasted in collisions by framed Aloha variants.

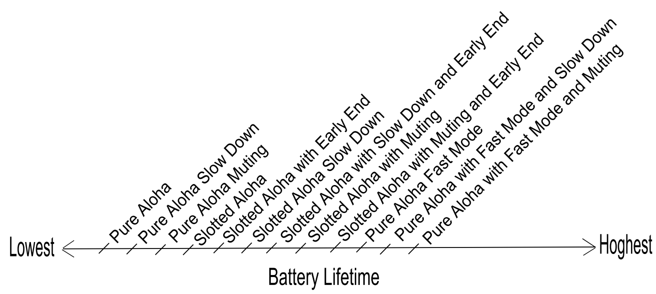


Fig. 25. Battery lifetime of pure/slotted Aloha variants.

ther have to re-read all tags again, or require expensive tree reconstruction. Hence, there is a clear need for energy efficient protocols that can determine new and old tags quickly. Moreover, such protocols must remain efficient in dynamic and high tag density scenarios. To date there is only one protocol, see [78], that have tackled both energy efficiency and monitoring simultaneously. Therefore, research into protocols that can track tags whilst remaining energy efficient is in its infancy.

Tag orientation affects the performance of tag reading protocols. In the worst case, if a tag's antenna is parallel to the reader's field lines, tags become unreadable [7]. This means when readers and tags are randomly deployed, there is a possibility that tags become unreadable even though they are in a reader's interrogation zone. An approach to overcome this problem is to develop cooperative tag reading protocols. In essence, we are interested in having multiple RFID reader equipped sensor nodes with overlapping interrogation zone cooperatively read a set of tags. The observation here is that given the number of deployed sensor nodes, it is likely that one of them will be better oriented to read tags that otherwise would be unreadable if there is only one reader. The analysis of such systems is nonexistent at this point in time.

Lastly, to the best of our knowledge, no works have carried out an analytical study on the performance of anti-collision protocols when tags are mobile. Hence, an interesting research work will be to construct a model that can determine the minimum time a mobile tag must remain in a reader's interrogation zone before it is identified.

VI. Conclusion

We have presented a comprehensive survey and classification of RFID anti-collision protocols. In general, two methods are used for identifying tags: Aloha and tree. The

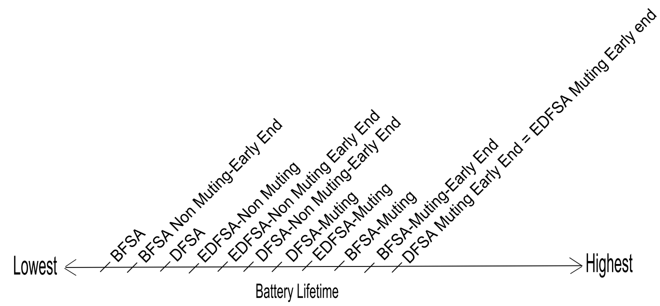


Fig. 26. Battery lifetime of framed Aloha variants.

key advantages of Aloha protocols are dynamic adaptability to varying loads and low reader to tag commands. On the other hand, tree protocols promise deterministic identifications, but require a high number of reader to tag commands.

Lastly, a key limitation of current RFID systems is the lack of multi-hop capabilities. A promising way to address this limitation is to create a RFID-enabled wireless ad-hoc network [83]. To this end, a promising research area is to develop protocols to coordinate the reading of tags by multiple readers. Readers are referred to [10][84] for further information on the reader collision problem, and its solutions.

VII. Acknowledgment

We acknowledge and thank the support of the Australia Research Council, grant number DP0559769.

References

[1] In-Stat, "Explosive growth projected in next five years for RFID tags." http://www.instat.com.

[2] RFID Journal, "Wal-mart begins RFID process changes." http://www.rfidjournal.com/article/articleview/1385.

[3] RFID Journal, "DoD releases final RFID policy." http://www.rfidjournal.com/article/articleview/1080/1/1.

[4] RFID Journal, "DoD reaffirms its RFID goals." http://www. rfidjournal.com/article/articleview/3211/1/1.

[5] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data and Computing*, pp. 322–333, 2006.

[6] R. Want, "The magic of RFID," *RFID: Threat or Promise? COLUMN: Q Focus: RFID*, vol. 2, no. 7, pp. 40–48, 2004.

[7] K. Finkenzeller, *RFID Handbook , Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley and Sons Ltd, 2003.

[8] S. Lahiri, *RFID Sourcebook*. USA: IBM Press, 2006.

[9] L. A. Burdet, "RFID multiple access methods." Technical Report. http://www.vs.inf.ethz.ch/edu/SS2004/DS/reports/06_rfid-mac_ report.pdf.

[10] D. H. Shih, P. L. Sun, D. C. Yen, and S. M. Huang, "Taxonomy and survey of RFID anti-collision protocols: Short survey," *in Computer Communications*, vol. 29, no. 11, pp. 2150–2166, 2006.

[11] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[12] S. Lewis, "Think RFID technology." http://hosteddocs.ittoolbox.com/laran032604.pdf, 2004.

[13] D. Paret, *RFID and Contactless Smart Card Applications*. UK: John Wiley and Sons, 2005.

[14] NXP Semiconductors, "HITAG." http://www.nxp.com/#/pip/cb=[type=product,path=/53420/53460/42027]—pip=[pfp=42027][0].

[15] EPIC, "EPIC questions to RFID industry." http://epic.org/privacy/rfid/survey.html.

[16] J. Mitsugi, "RFID technology." http://www.apricot.net/apricot2005/slides/KT2_3.pdf.

[17] IDTechEX, "Near field UHF vs HF." http://www.idtechex.com/products/en/articles/00000474.asp.

[18] PowerID, "The advantage of battery-assisted UHF RFID labels." White paper. http://www.power-id.com/fileserver.php?file=20 .

[19] IDTechEX, "Chipless RFID tags." http://www.idtechex.com/products/en/articles/00000435.asp.

[20] Z. Tang and Y. He, "Research of multi-access and anti-collision protocols in RFID systems," in *IEEE International Workshop on Anti-counterfeiting, Security, Identification*, (Xiamen, China), pp. 377–380, 2007.

[21] A. Rohatgi and G. D.Durgin, "RFID anti-collision system using the spread spectrum technique," in *Technical Report*, pp. 1–19, 2005. http://www.propagation.gatech.edu/Archive/PG_TR_050425_AR/PG_TR_050425_AR.pdf.

[22] E. Microelectronics, "Supertag category protocols." Datasheet. http://www.gaw.ru/doc/EM-Marin/P4022.PDF.

[23] D. K. Klair, K.-W. Chin, and R. Raad, "An investigation into the energy efficiency of pure and slotted Aloha based RFID anti-collision protocols.," in *In IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM'07)*, (Helsinki, Finland), IEEE, 2007.

[24] M. Schwartz, *Telecommunication Networks Protocols, Modeling and Analysis*. USA: Addison-Wesley, 1988.

[25] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed Aloha for multiple RFID objects identification," *IEICE-Transactions on Communications*, vol. E88-B, pp. 991–999, 2005.

[26] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *The 11th Intl. Conference on Parallel and Distributed Systems*, (Korea), pp. 63–67, 2005.

[27] T.-W. Hwang, B.-G. Lee, Y. S. Kim, D. Y. Suh, and J. S. Kim, "Improved anti-collision scheme for high speed identification in RFID system," in *Proc. First International Conference on Innovative Computing, Information and Control*, (China), pp. 449–452, 2006.

[28] J. Wang, Y. Zhao, and D. Wang, "A novel fast anti-collision algorithm for RFID systems," in *International Conference on in Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, (Shanghai, China), 2007.

[29] H. Vogt, "Multiple object identification with passive RFID tags," in *The IEEE Intl. Conf. on Man and Cybernetics*, (Tunisia), pp. 6–13, 2002.

[30] H. Vogt, "Efficient object identification with passive RFID tags," in *IEEE PerCom*, (TX, USA), 2002.

[31] G. Khandelwal, A. Yener, K. Lee, and S. Serbetli, "ASAP: a MAC protocol for dense and time constrained RFID systems," in *IEEE International Conference on Communications (ICC'06)*, (Istanbul, Turkey), 2006.

[32] C. Floerkemeier and M. Wille, "Comparison of transmission schemes for framed Aloha based RFID protocols," in *Proc. International Symposium on Applications on Internet Workshops*, (Phoenix, AZ, USA), 2006.

[33] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in *The 4th Annual Intl. Conference on Pervasive Computing and Communications Workshops*, (Pisa, Italy), 2006.

[34] R. Rivest, "Network control by bayesian broadcast," *IEEE Trans. Inf. Theory*, vol. IT-33, pp. 323–328, May 1987.

[35] W.-T. Chen and G.-H. Lin, "An efficient anti-collision method for RFID system," *in IEICE Trans. Commun.*, vol. E89, no. B, pp. 3386–3392, 2006.

[36] W.Feller, *An Introduction to Probability Theory and its Applications*. New York: Addison-Wesley, 1970.

[37] A. Technology, "EPCglobal class 1 gen 2 RFID specifications." Whitepaper. http://www.alientechnology.com/docs/AT_wp_EPCGlobal_WEB.pdf.

[38] D. Klair, K.-W. Chin, and R. Raad, "On the Accuracy of Tag Estimation Functions," in *Sixth IEEE International Symposium on Communications and Information Technologies*, (Sydney, Australia), 2007.

[39] Auto-ID Center, "13.56 MHz ISM band class 1 radio frequency identification tag interface specification, version 1.0." HF RFID standard.

[40] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted Aloha algorithm for RFID tag identification," in *The 2nd Intl. Annual Conference on Mobile and Ubiquitous Systems: Networking and Services*, (San Diego, USA), pp. 166–172, 2005.

[41] D. K. Klair, K.-W. Chin, and R. Raad, "On the suitability of framed Aloha based RFID anti-collision protocols for RFID-Enhanced WSNs," in *The 16th IEEE International Conference on Computer Communications and Networks (IEEE ICCCN'2007)*, (Honolulu, Hawaii, USA), 2007.

[42] D. Klair, K.-W. Chin, and R. Raad, "On the energy consumption of pure and slotted Aloha anti-collision protocols in RFID-enhanced wireless sensor nodes," in *Elsevier Computer Communications.*, 2009. In Press.

[43] F. C. Schoute, "Dynamic frame length Aloha," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, 1983.

[44] C. J, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 505–515, 1979.

[45] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *The IEEE Intl. Symposium on Information Theory*, (Mexico City, USA), pp. 107–107, 1998.

[46] J. Myung and W. Lee, "Adaptive binary splitting: A RFID tag collision arbitration protocol for tag identification," in *IEEE BROADNETs*, (Boston, MA, USA), pp. 347–355, Oct. 2005.

[47] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Personal Commun. Mag.*, vol. 10, no. 3, pp. 144–146, 2006.

[48] W.-C. Chen, S.-J. Horng, and P. Fan, "An enhanced anti-collision algorithm in RFID based on counter and stack," in *Second International Conference on in Systems and Networks Communications, (ICSNC)*, (Cap Esterel, French Riveira, France), pp. 21–24, Aug. 2007.

[49] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification (extended abstract)," in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, (Toronto, CA), pp. 75–84, Aug. 2000.

[50] J. H. Choi, D. Lee, Y. Youn, H. Jeon, and H. Lee, "Scanning based pre-processing for enhanced RFID tag anti collision protocols," in *International Symposium on Communications and Information Technologies (ISCIT)*, (Bangkok, Thailand), 2006.

[51] J. Myung, W. Lee, and T. Shih, "An adaptive memoryless protocol for RFID tag collision arbitration," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1096–1101, 2006.

[52] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocol for RFID networks," in *the 24th IEEE Annual Conference on Computer Communications (INFOCOM 2005)*, (Miami, USA), Mar. 2005.

[53] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *Proc. 2004 international symposium on Low power electronics and design*, pp. 357–362, 2004.

[54] J. H. Choi, D. Lee, and H. Lee, "Query tree-based reservation for efficient RFID tag anti-collision," *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 85–87, 2007.

[55] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Randomized hashing for tag identification in RFID networks," in *Technical Report: Computer Communications and Networks*, 2005.

[56] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent query tree (IQT) protocol to improve RFID tag read efficiency," in *9th International Conference on Information Technology (ICIT'06)*, pp. 46–51, 2006.

[57] S. Yu, Y. Zhan, Z. Wang, and Z. Tang, "Anti-collision algorithm based on jumping and dynamic searching and its analysis," *Computer Engineering*, vol. 31, pp. 19–20, 2005.

[58] L. Liu, Z. Xie, J. Xi, and S. Lai, "An improved anti-collision algorithm in RFID aystem," in *2nd International Conference on Mobile Technology, Applications, and Systems*, (Bangkok, Thailand), pp. 1–5, Oct. 2005.

[59] F. Bo, L. Jin-Tao, G. Jun-Bo, and D. Zhen-Hua, "ID-binary tree stack anticollision algorithm for RFID," in *Proc. 11th IEEE Symposium on Computers and Communications*, (Sardinia, Italy), pp. 207–212, Apr. 2006.

[60] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contact-less identification device with anti-collision algorithm," in *Conference on Circuits Systems, Computers and Communications*, (Athens, Greece), July 1999.

[61] H.-S. Choi, J.-R. Cha, and J.-H. Kim, "Fast wireless anti-collision algorithm in ubiquitous ID system," in *Vehicular Technology Conference*, pp. 4589–4592, 2004.

[62] S. H. Kim, M. K. Shin, and P. Park, "A new tree-based tag anti-collision protocol for RFID systems," in *Intl. Conf. on Communications in Computing*, (Nevada, USA), pp. 83–86, June 2006.

[63] S. H. Kim and P. Park, "An efficient tree-based tag anti-collision protocol for RFID systems," *IEEE Commun. Lett.*, vol. 11, pp. 449–451, 2007.

[64] Chris Turner, "Deterministic or probabilistic which is best?." Weblink. //www.rfipsolutions.com/downloads/Deterministic_or_Probabilistic.pdf.

[65] Chris Turner, "Evaluating the merits of various RFID tagging protocols with respect to the protocol chracteristics." Technical Report. http://rfip.eu/paper.php?pid=tech:9.

[66] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree slotted Aloha: a new protocol for tag identification in RFID networks," in *International Symposium on on World of Wireless, Mobile and Multimedia Networks*, (Niagara-Falls, USA), pp. 603–608, 2006.

[67] J. Ryu, H. Lee, Y. Seok, T. Kwon, and Y. Choi, "A hybrid query tree protocol for tag collision arbitration in RFID systems," in *IEEE ICC*, (Scotland), pp. 5981–5986, June 2007.

[68] J.-D. Shin, S.-S. Yeo, T.-H. Kim, and S. K. Kim, *Hybrid Tag Anti-collision Algorithms in RFID Systems*. UK: Springer Berlin / Heidelberg, 2007.

[69] V. Namboodiri and L. Gao, "Energy-aware tag anti-collision protocols for RFID systems," in *Fifth Annual IEEE International Conferance on Pervasive Computing and Communications (PerCom)*, (NY, USA), pp. 23–46, 2007.

[70] H. Zhang, L. Han, and Y.-L. Li, "Design of hash-tree anti-collision algorithm," in *Third International Conference on Natural Computation (ICNC)*, (China), pp. 176–179, 2007.

[71] EPCglobal, "Electronic product code." http://www.epcglobalinc.org/home/.

[72] ISO, "RFID Standards." http://www.iso.org/iso/search.htm?qt=RFID &published=on&active_tab=standards.

[73] Jae-Hyun Kim, "Multi-reader/multi-tag anti-collision." www.krnet.or.kr/board/include/download.asp?no=3&db=program&fileno=2.

[74] D. Baddeley, "ISO/IEC FCD 14443-3." http://www.waazaa.org/download/fcd-14443-3.pdf.

[75] OTI, "ISO 14443." http://www.otiglobal.com/objects /ISO%2014443%20WP%204.11.pdf.

[76] NXP, "U CODE." http://en-origin.nxp.com/ #/pip/pip=[pfp=50483]—pp=[t=pfp,i=50483].

[77] NXP, "MIFARE." http://en-origin.nxp.com/#/homepage /cb=[t=p,p=/53420/53422]—pp=[t=pfp,i=53422].

[78] D. Klair and K.-W. Chin, "A novel anti-collision protocol for energy efficient identification and monitoring in RFID-enhanced wsns," in *The 17th IEEE International Conference on Computer Communications and Networks (IEEE ICCCN'2008)*, (St Thomas, US Virgin Islands, USA), IEEE, August 2008.

[79] L. Ho, M. Moh, Z. Walker, T. Hamada, and C.-F. Su, "A prototype on RFID and sensor networks for elder healthcare: Progress report," in *Proceeding of the 2005 ACM SIGCOMMWorkshop on Experimental Approaches to Wireless Network Design and Analysis*, (Philadelphia, Pennsylvania, USA), pp. 70–75, ACM Press, 2005.

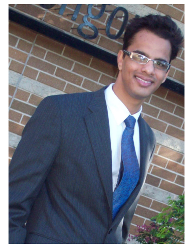
[80] Intel, "Intel sensor nets/RFID research areas." Webpage. http://www.intel.com/research/exploratory /wireless_sensors.htm.

[81] Jonathan Collins, "NASA creates thinking RF sensors." http://www.sensorwaresystems.com/historical/press/RFIDJrnl_NASA _Oct04.pdf.

[82] M. Roberti, "BP leads the way on sensors." Webpage. http://www.rfidjournal.com/article/articleview/1216/1/2/.

[83] D. Klair, K.-W. Chin, R. Raad, and D. Lowe, "A spatially aware RFID-enhanced wireless sensor networks," in *The sixth IEEE International Conference on Pervasive Computing and Communications (IEEE PerCom): Google PhD Forum*, (Hong Kong, China), IEEE, march 2008.

[84] G. P. Joshi and S. W. Kim, "Survey, Nomenclature and Comparison of Reader Anti-Collision Protocols in RFID," *IETE Technical Review*, vol. 25, no. 5, pp. 285–292, 2008.



Dr Dheeraj K Klair obtained his PhD from the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Australia in July 2009. His thesis entitled "On the suitability of RFID anti-collision protocols in energy constrained environments" was awarded with the examiners' commendation for outstanding PhD thesis. In June 2009, he joined the Smart Services CRC and Information and Communication Technology Research Institute at the University of Wollongong. His current research interests include digital image and video processing, general-purpose computation on graphics hardware, medium access control protocols for RFID systems, wireless sensor networks and RFID-enhanced wireless sensor networks.

Dr Kwan-Wu Chin obtained his Bachelor of Science with first class Honours from the Curtin University of Technology, Australia. He then pursued his PhD at the same university, where he graduated with distinction and the vice-chancellors commendation. After obtaining his PhD, he joined Motorola Research Lab as a senior research engineer, where he developed zero-configuration home networking protocols and designed new medium access control protocols for wireless sensor networks. In 2004 he joined the University of Wollongong as a senior lecturer. His current research areas include medium access control protocols for wireless networks, routing protocols for delay tolerant networks, RFID anti-collision protocols, and the application of network coding in wireless networks. To date, he holds four US patents, and has published more than 35 articles in numerous conferences and journals.

Dr Raad Raad is a lecturer at the University of Wollongong where he has been there since 2004. Previously he held the position of Staff Research Engineer at Motorola Australia Research Labs, Sydney, Australia between 2001 and 2004. He has been awarded a number of patents by the US patent office for his work on wireless sensor networks and IEEE 802.11 WLAN. He has conducted research into a number of areas that include admission control, Fuzzy Logic, Bandwidth Management, wireless sensor networks, and RFIDs.