RFID ANTI-COLLISION TECHNIQUE: COHERENT COLLISION

BY

RONALD J. ROTH

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Advisers:
Professor Bruce Hajek
Professor Kenneth Loparo, Case Western Reserve University
RFID technology has enabled great advances in the asset tracking industry. Shipping, warehouse inventory, and storefront security are just a few examples where the application of RFID systems has increased efficiency and lowered cost. A major area of research at the present time is concerned with optimal ways of reading multiple ID tags using a single reader as efficiently as possible. These methods are called "Anti-collision" protocols, and they all seek to somehow arbitrate how a multitude of ID tags and a reader negotiate the process of reading all of the ID tags. As the name implies, nearly all of the methods seek to detect and avoid collisions between ID tags, therefore reading one ID tag at a time. This thesis seeks to present another, novel approach to solving the same problem. The proposed method seeks not to avoid collisions, but rather to orchestrate them in a manner that allows overlapping transmission of the ID tags while identifying all of the ID tags that take part in the transmission. This method will provide a highly time-efficient collision management protocol.
I would like to acknowledge the contributions to the fruition of this thesis that came from many people, both directly and indirectly. I would like to thank my advisers, Professor Hajek and Professor Loparo, both of whom worked closely with me to help guide my work. Additionally, Professor Eden was instrumental in helping me grow while studying in the courses taught by him and working in his lab as a graduate student.

I would also like to thank my employer for allowing me the flexibility to take time to work on completing my degree. The magnitude of time and effort involved would have made it prohibitive to complete this without their support.

I would like to thank the members of my family who helped make this possible. My mother, father, and grandmother are central to my success in nearly everything I do. They provided me with a stable, warm, and loving home that valued learning, creativity, and growth. That environment allowed me to thrive intellectually. Without their strength, wisdom, and support, I would not have been able to complete this course of study or succeed in many other areas of life. For that, I am forever grateful.

My wife has helped me tremendously by supporting and encouraging me throughout the course of working on this thesis. She worked to ensure that I had time to focus on my work, while taking on a greater burden at home. Her love, patience, and support were extremely valuable to me. For that, I am forever blessed.

Finally, my young children, you are too young to understand how you contributed. But you did, and always in the cutest way possible. You motivate me to do things that I would otherwise have given up on. For that, I hope to forever be a blessing for you.
# TABLE OF CONTENTS

Chapter 1: Introduction .................................................................................................................... 1  
 1.1 A Brief History of Asset Management .................................................................................. 1

Chapter 2: Background .................................................................................................................. 4  
 2.1 Basic RFID System Topology and Operation ........................................................................ 4  
 2.2 Existing Challenges and Design Considerations ..................................................................... 7  
 2.3 RFID Communication Basics ............................................................................................... 9

Chapter 3: Proposed Algorithm .................................................................................................. 26  
 3.1 Protocol Description .............................................................................................................. 28  
 3.2 Terminology .......................................................................................................................... 29  
 3.3 Definition of ID Tag States .................................................................................................... 30  
 3.4 System Initialization .............................................................................................................. 32  
 3.5 Synchronization Pulse ........................................................................................................... 32  
 3.6 First ID Arbitration ................................................................................................................ 33  
 3.7 Cluster Transmit .................................................................................................................... 34  
 3.8 Return to Synchronization .................................................................................................... 37  
 3.9 Reader-Feedback Mechanism ............................................................................................... 37  
 3.10 ID Tag Identification ........................................................................................................... 38  
 3.11 Example ............................................................................................................................... 40

Chapter 4: Performance Analysis of Proposed Algorithm .......................................................... 46  
 4.1 Experimental Methodology ................................................................................................... 46  
 4.2 Simulation Methodology ....................................................................................................... 48  
 4.3 Model Parameters and Assumptions ..................................................................................... 49  
 4.4 Baseline Comparison ............................................................................................................ 49  
 4.5 Comparative Bounds and Strategies ..................................................................................... 50  
 4.6 Results .................................................................................................................................. 54

Chapter 5: Conclusions ................................................................................................................. 62

References ....................................................................................................................................... 64
Chapter 1: Introduction

Radio Frequency Identification (RFID) is a tool that, at its most basic level, is concerned with transmitting the identifying information contained in an ID tag, which can then be translated into some user-imposed meaning. One of the more common user-imposed meanings is to use the ID as a tag (hence the term RFID-tag or ID tag) which is physically tied to a piece of inventory or an asset. In this way, reading the RFID-tag can uniquely identify the presence of a given asset. For example, RFID tags have gotten so small that they can be safely implanted inside living beings, such as wildlife or livestock, and then tracked for the remainder of their lives [1, Section 2.2.3]. Pets can conceivably be tracked in a similar manner to identify them when they go astray and perhaps show up at an animal shelter. The RFID tag is also used as an industrial tool to track inventory in a warehouse, or during shipping [2].

1.1 A Brief History of Asset Management

The need to track assets has been around for as long as humans have had a concept of ownership. Ancient cultures used complex systems of tracking assets. These systems involved using written logs or other representations of, for example, how much gold a ruler possessed or how much food was produced in a given season. Farmers have been known to brand their cattle in order to identify it as their own [3]. As the scale of human activity increased throughout history, the need to track these assets on a larger scale became even more complex. New, innovative solutions were generated with each new challenge. More recently, optically-scanned barcodes were created to enable a fully-integrated computer-based tracking system which could identify assets [1]. These
systems represent a significant improvement over previously existing methods – but certain limitations existed that left room for improvement. The most obvious is the speed at which barcodes could be read – each barcode had to be individually scanned by an employee (or a machine, in some specialized cases), which required that the barcode be physically moved in order to present it to the barcode reader. When dealing with a stack of boxes, or a bin full of a multitude of items of interest, this may be very time-consuming and inefficient. Many people have likely wished that the line at the grocery store could move faster – and most of the time there is spent scanning barcodes.

The advent of RFID systems allowed two immediate advances in efficiency – one was that the RFID reader only requires that the RFID tag be within the vicinity of the reader, so the inventory does not need to be physically moved to present it to the reader in order to identify it. The other is that a single reader can read all of the nearby RFID tags in a short time, with a single push of a button.

This thesis will examine the communication medium management problem that is encountered when attempting to identify a multitude of ID tags which are within reading range of the reader. The techniques used to solve this problem are broadly referred to as “anti-collision” protocols. Existing techniques are functional, but leave considerable room for improvement in their efficiency. This thesis will explore the short-comings of current anti-collision protocols and then propose an alternate protocol that greatly improves the efficiency of the communication (by orders of magnitude in some conditions). It performs particularly well in densely populated fields, which is a region of operation that some existing protocols do not deal with very well.
This thesis is organized as follows: Chapter 1 (this chapter) contains a brief introduction to RFID uses and provides the motivation for pursuing the ideas presented in this thesis. Chapter 2 provides background on the basic operational aspects of a generic RFID system, as well as the basic underlying concept of operation. It also gives treatment to the existing challenges facing RFID technology. Chapter 2 concludes with a more detailed background on RFID communication, including detail on the operation of the readers and the tags in an RFID system. It also provides a survey of existing collision management (anti-collision) protocols. Chapter 3 provides a detailed exposition regarding the proposed algorithm. Chapter 4 is concerned with a performance analysis of the proposed algorithm. It covers details regarding the simulation method used, performance metrics considered, as well as the results of the proposed algorithms. The chapter includes a summary and discussion of the results, with comparison to the established results of an existing protocol used as a baseline comparison. Chapter 5 provides conclusions and proposed next steps that the research in this field may take.
Chapter 2: **Background**

The wireless communication basics of an RFID system are covered here, as well as a survey of existing anti-collision protocols. There are many variants of RFID systems. In the interest of clarity, this thesis will only discuss the RFID systems for which the proposed algorithm is intended (although it could, conceivably, be adapted for other types of RFID systems).

The type of systems considered in this thesis are inductively coupled, full or half-duplex, with passive RFID tags that use load modulation to transmit data from the RFID tag to the reader. For more background regarding these systems, the reader is referred to section 3.2.1 in [1].

### 2.1 Basic RFID System Topology and Operation

An RFID system has two main categories of components – readers and tags. The reader is responsible for identifying the tags that are present within its read-field. The tags are responsible for permanently storing their identity information, and relaying that information to the reader when appropriate. Some RFID systems are designed with some additional functions or a more complex topology.

The basic RFID system topology considered in this thesis is shown in Illustration 2.1.

---

1 Some RFID systems contain protocols for overwriting the data within a given tag [1], [4]. This thesis does not concern itself with that topic. Additionally, some RFID systems are designed to have multiple RFID readers with overlapping read-fields, and therefore require collision management protocols for the readers as well as the tags [5]. This thesis does not concern itself with that scenario either, but using this protocol or something similar for that problem may become a direction of future research.
Many RFID systems exist, and they differ in their implementation details, but the basic operation occurs as follows (see section 3.2.1.2 of [1] for details): The RFID reader sends out a signal which activates the ID tags and informs them that they are going to be interrogated for their stored ID information. The ID tags respond by transmitting their ID, and the reader is responsible for correctly identifying which ID tags are present. The reader is also responsible for identifying and managing collisions.

A typical arrangement consists of a reader which activates the ID tags by producing a carrier wave of a specified frequency and power [1, Section 3.2.1.1]. The ID tags have antennas which are tuned to receive that frequency. The ID tags use the received power of the carrier wave to power their internal circuitry. When the ID tag is ready to transmit its information, it does so by closing and opening a loading circuit attached to the antenna using an internal switch according to the bit-sequence that defines its...
internally stored data [1, Section 3.2.1.2]. When the loading circuit is closed, the inductive coupling between the reader's transmitting antenna and the ID tag's receiving antenna produces a small voltage drop in the reader's antenna. The reader can thus identify the bits in the ID tag's bit-sequence by detecting the presence or absence of this voltage drop. Illustration 2.2 shows the operation of this type of a system. This type of system is the primary focus of this thesis – although the proposed algorithm can potentially be applied to other types of systems.
2.2 Existing Challenges and Design Considerations

Implementing an RFID system requires solving some challenges that are common to nearly all applications. These challenges can be generically viewed as finding an optimal balance between certain system performance metrics (time and power efficiency, accuracy) and system costs (costs of the reader and ID tags).

2.2.1 Collisions

The first and most obvious challenge is that of handling the case in which more than one ID tag attempts to transmit its ID at the same time. This is termed a “collision”. The problem is shown pictorially in Illustration 2.3. Most ID tags are unaware of the activity of any other ID tags due to the fact that ID tags must be made small and cheap for most applications. Therefore, they cannot actively avoid transmitting at the same time as other tags by checking if the medium is busy, as is done in many other communication protocols [6]. The figure illustrates the effect of transmissions at the same time from two ID tags. Due to the fact that the ID tags are unaware of each other, their transmissions are generally not synchronized, but the reader is likely synchronized with at least one of them. Without loss of generality, the illustration assumes that the reader is synchronized with ID Tag 1. The reader will experience signal level transitions in the middle of a bit-time, leading to an unreliable decode (signified by the “???” in the “Decoded Symbol” row), so the reader must declare that it has experienced a collision. The result is that the reader is unable to identify either tag, and both ID tags must be queried again later. The reader may also use a CRC checksum or similar embedded error-detection to detect a collision, since a synchronized transmission of multiple tags...
may result in an apparently coherent message that can be decoded to a binary number, but may not actually represent one of the ID tags in the read-field [1, Section 7.2.4.3].

Illustration 2.3: RFID Collision

2.2.2 Throughput

The speed at which ID tags can be identified, especially a multitude of ID tags in the same read-field, is called throughput. Many factors impact throughput, including the baud rate, collision-management protocol, number of tags in the read-field, etc. Throughput is an important metric for evaluating the efficiency of the RFID system, and designing a good anti-collision protocol is of paramount importance for throughput. The most widely used protocols achieve less than 50% throughput today [6].
2.2.3 Tag Complexity

One of the major costs in an RFID system that is intended to be used in an industrial inventory tracking system is that of the ID tags. They must be exceedingly cheap and simple in order to facilitate mass production without making the entire system cost-prohibitive. They must also have a simple and compact construction in order to allow them to be used even when the item being tracked is not large. Additionally, the ID tags should be considered disposable for maximum flexibility. The reader, on the other hand, may be more complex because there will be less of them, but still cannot be prohibitively expensive. Additionally, if the reader will be used by employees daily (for example, in the package delivery industry), the reader should meet certain ergonomic restrictions.

The desire for low-cost tags means that many other communication-media management protocols, such as those used for computer connectivity like Ethernet, cannot be used because the protocol usually requires a microprocessor (which is power-hungry when the only source of power is received radiation on a small antenna).

2.3 RFID Communication Basics

2.3.1 Wireless Communication

RFID systems must operate using a shared wireless communication medium. This section provides a basic understanding of the operation of the components of an RFID system (Readers and Tags), and how they communicate with each other. Some of this material was touched on in Chapter 2: “Background”, but will be explored further here.
2.3.1.1 Reader Construction and Operation

The reader contains a processor, some memory storage, an antenna-driving circuit and an antenna-reading circuit. The reader also contains or is connected to some sort of a power source. When requested, the reader will initiate a sequence of operations by which it will attempt to identify all of the ID tags within its read-field. The reader will generate a carrier wave at a specific frequency in order establish a read-field. The presence of the read-field provides energy to power the RFID tags and also provides a communication medium from which the RFID tags can read data [1, Section 3.2.1].

The reader will then, in some prescribed way, inform the ID tags that they should start transmitting their IDs. In some systems, the reader will generate a synchronization pulse by briefly shutting off the carrier wave for a short duration – long enough to be easily detected by the tags, but short enough that the tags will not become de-energized [4]. In other systems, the tags simply have a random time delay from the moment they become energized until they begin transmitting [6]. Still other systems will use some sort of a preamble that is encoded in the carrier wave to tell specific tags (with a matching preamble) that it is time to transmit [6].

The reader will detect the tag responses (see detail in the section titled “Detection of Tag Transmissions” below) and determine if a collision occurred or if the read was successful in identifying a particular tag. If a collision occurred, then the reader will take appropriate measures to deal with it. See the section “RFID Anti-Collision Protocols” below.
A functional block diagram of a typical RFID reader is shown in Illustration 2.4 for reference [1, Section 4.1.10].

![Illustration 2.4: RFID Reader Functional Block Diagram]

### 2.3.1.1.1 Carrier Frequency Generation

The reader must generate a carrier frequency at a specified operating frequency. Most readers generate only one or two frequencies, but the range of frequencies used in various implementations goes from about 9 kHz all the way up to several GHz. An inductively coupled system, such as the type considered in this thesis, will typically use a carrier field frequency of about 100 kHz to about 13 MHz [1], [6].

The reader will use an inductive coil, typically round or rectangular, as the radiating element [1, Chapter 2]. As shown in Illustration 2.4 above, a high-frequency signal (perhaps from a sine-wave oscillator or square wave generator) is used to generate a pre-amplification voltage signal that contains the desired carrier frequency. The voltage signal is then amplified to a higher voltage and buffered to have a higher current-driving capacity. The resulting signal drives the series-resonant circuit formed by C1 and L1. The components L1 and C1 form a series resonance circuit, and their values are
chosen such that their resonant frequency is the desired transmission frequency. The resonance causes the circuit to build very high oscillating currents, which allows the antenna to generate the field [1, Section 4.1.10].

2.3.1.1.2 Detection of Tag Transmissions

The reader is responsible for detecting the transmissions that are generated by the ID tags. The system under consideration uses a technique called “load modulation” for purposes of ID tag → reader data transmission (see section on “Load Modulation” below).

The presence of the receiving antenna in the field generated by the reader creates a mutual coupling between the two antennas. The field will create a voltage in the receiving antenna, which acts as a load on the transmitting antenna. This load can be seen as a “virtual impedance” as though it was an actual component added in series with the receiving antenna and having a particular impedance. As long as the receiving antenna is present in the field and part of a complete circuit, it will create a small additional load on the transmitting antenna. Note that even merely powering the ID tag's circuitry will act as a load [1, Section 4.1.10.1].

Load modulation works by allowing the ID tag to change the magnitude of its loading (see details in the section on “Load Modulation” below). This change in loading means that the additional impedance that is induced by the loading from the ID tag will also be changed, or modulated. This is often accomplished by adding a loading resistor in parallel with the remainder of the ID tag's circuitry (as shown in Illustration 2.2 above) [1, Section 4.1.10.3]. The additional parallel resistance results in a lower impedance across
the ID tag’s coil, and therefore reduces the virtual impedance created in the transmitter's coil [1, Section 4.1.10.1].

This reduced impedance in the transmitter’s antenna will mean that the voltage will also be reduced – thus creating an amplitude modulation of the voltage across the antenna coil. The bits can then be decoded with typical AM demodulation techniques, such as an envelope detector with a threshold detector, or a product detector [1, Section 4.1.10.3].

2.3.1.2 Tag Construction and Operation

The ID tags contain a receiving antenna, some simple logic circuitry (i.e. for clocking out the stored data for transmission), and a power circuit that uses energy from the receiving antenna to power the rest of the ID tag circuitry. The ID tag, in addition to its functional requirements, has a strong requirement to be low cost and simple to produce in mass quantities. A functional block diagram of the ID tag is shown in Illustration 2.5 [1, Section 4.1.8].
2.3.1.2.1 Wireless Power

The ID tags have an antenna circuit which has a resonance at the same frequency as the carrier field generated by the reader (by properly selecting the values of L1 and C1). This resonance generates currents within the receiving antenna when excited by the carrier field. A rectifying circuit and capacitor can be used to store the captured energy for use by the remaining circuitry, as shown in Illustration 2.6. Additional considerations (not shown) are required to limit the voltage that appears across C2 in order to protect the circuitry from the high voltages that can be generated in the antenna's resonant circuit [1, Sections 3.2.1 and 4.1.8.2].
2.3.1.2.2 Load Modulation

The ID tag uses load modulation to transmit its information back to the reader. The way this is accomplished is to somehow change the loading on the ID tag's antenna circuit, thereby inducing a change in the transmitter's antenna (see previous section regarding “Detection of Tag Transmissions”). Two very common methods are to switch a capacitor in parallel with the antenna circuit, thereby detuning it from the carrier wave frequency, or to switch a resistor in parallel with the antenna circuit. A diagram is shown in Illustration 2.7 [1, Section 4.1.10.3].
This section provides a brief overview of existing RFID anti-collision protocols. Note that there are many variations of these protocols – only enough explanation is given here to sufficiently illustrate the general operation of the protocol. The performance results presented here are published results, not simulation results generated by the author.

### 2.3.2.1 ALOHA

The ALOHA protocol is so named for its origin – the protocol was developed for ALOHANET, which was developed by the University of Hawaii in the 1960s, so that wireless data packets could be transmitted between stations participating in the network [1, Section 7.2.4.1].

#### 2.3.2.1.1 Protocol Description

The ALOHA protocol has many variants. The sections below cover the most relevant ones – Pure ALOHA, Slotted ALOHA, and Framed Slotted ALOHA.
2.3.2.1.1 Pure ALOHA

The “original” protocol is referred to as “pure” ALOHA. This protocol can be described very simply as having two rules:

- If a node is ready to send data, then send immediately.
- If a collision occurs during transmission, cease transmitting and then re-try again after a random time delay.

In the original implementation on ALOHANET, each node would become ready to send information in a random manner, depending on user activity. Adapting this same protocol to RFID systems required that tags always begin with a random time delay before attempting their first transmission. Otherwise, the first transmission would be guaranteed to result in a collision anytime that there is more than one tag present. This protocol is commonly depicted pictorially as shown in Illustration 2.8.

An ID tag is only aware of a collision at the end of its transmission. This is because the reader will send an ACK signal (as a predefined modulation of the carrier field, such as a brief pulse during which the carrier field is switched off) when it has successfully received an IDTBS without collision. If the tag does not receive the ACK, then it assumes that a collision has occurred. This means the tag is transmitting for an entire slot (i.e. a period of time wide enough to transmit an entire ID tag bit-sequence) before it
can learn if the reader has received its information. In Pure ALOHA, another tag may interrupt the first tag at any point during the transmission [1], [6].

2.3.2.1.2 Slotted ALOHA

In order to reduce the likelihood of a collision, the “slotted” ALOHA protocol was introduced [1], [6]. In this protocol, ID tags still randomize their start-of-transmission times, but they may only start at the beginning of a “slot”. The slot times start at synchronized points on the time axis, which means that if a collision were to occur, it would only occur during that slot and end when the slot is over – but would not continue into the next slot. If a collision occurs, the collided ID tags choose a random number of slots to wait before retransmitting. A common depiction of the slotted ALOHA protocol is shown in Illustration 2.9.

Illustration 2.9: Slotted ALOHA Protocol

2.3.2.1.3 Framed Slotted ALOHA

Even with a slotted ALOHA system, it is possible that a populous ID tag set or a particularly zealous ID tag (perhaps because its randomly selected wait times tend to be low) which tries to transmit often will result in a large number of collisions, reducing performance. This can be dealt with by adding a longer time-step synchronization on top of the slots used in slotted ALOHA. This is called a “Frame”, and the protocol is
termed “Framed Slotted ALOHA” [1], [6]. The frame consists of a number of slots (which can be either constant, or, as in a particularly interesting variant called Dynamic Framed Slotted ALOHA, varied in length as a function of the estimated number of tags that need to be identified), and ID tags are required to attempt transmission exactly once during the frame. This prevents a few over-zealous tags from drowning out other tags, or a few “shy” tags from extending the time required to identify them beyond what it would be otherwise. This protocol is depicted in Illustration 2.10. The Dynamic Framed Slotted ALOHA variant is the highest performing variant of all the ALOHA protocols [6].

Illustration 2.10: Framed Slotted ALOHA

2.3.2.1.2 Criticism

The ALOHA protocol should be praised for its simplicity. However, the highest performing variant, the DFSA, depends strongly on selecting a tag-population estimation function that suits the application [7], [8]. There are a myriad of proposed tag-population estimation functions [6], [8]. Clearly, selecting the right or wrong estimation function (for a given application) can have an impact on performance.

ALOHA also suffers from a peak performance throughput that is somewhat low. Only one tag can successfully transmit in a given slot, and less than half of the available slots
typically result in a successful transmission. This leaves a fair amount of room for improvement.

2.3.2.2 Tree-Based

Tree-based protocols are a family of protocols that operate on the principle that the reader will use a series of decisions to split the ID tag population into subsets for identification. Repeated application of these decisions, but with slightly different choices at each decision (i.e. a different path in the “tree”, hence “tree-based”) will allow the reader to reach a different tag [6].

2.3.2.2.1 Protocol Description

The major sub-families of tree-based protocols are briefly discussed below. Each sub-family relies on a slightly different decision mechanism, and each sub-family has many variants which optimize the protocol using various methods [6].

Some of the features of tree-based protocols are particularly important to the topic of this thesis, since they are used in the proposed algorithm. Specifically, the “bit-level synchronization” (found in the “bit-wise arbitration” algorithm [6]) and “prefix matching” (found in the “query tree” algorithm [6]) are relevant.

2.3.2.2.1.1 Tree Splitting

In a Tree-Splitting algorithm, the identification process and collision detection are performed in a similar manner as in the ALOHA protocols. Once a collision is detected, the reader informs the tags (through some reader-feedback mechanism) that a collision
has occurred. The ID tags then generate a random number and store the value in a
count-down register. This random number is similar to the random time-delay used in
the ALOHA protocol, but the register is modified after each time-slot based on whether
or not a collision occurred. If a collision did not occur, then the counter is decremented
by one. Otherwise, the tag generates another random number and adds it to the count-
down register, thereby increasing its time delay before transmission. When the count-
down register reached zero, the tag transmits [6].

The “splitting” mechanism used in this protocol is to split the population based on the
random numbers generated by the tags. The reader does not have knowledge of the
individual ID tag's random number or count-down register states.

2.3.2.1.2 Query Tree

This protocol uses a feature called “Prefix Matching” which is also an important feature
of the proposed algorithm. The reader will transmit a series of bits to the ID tags and
the ID tags will compare the bits to their prefix. The prefix may be part of the ID tag's ID
bit-sequence, or it may be another independent number. If the ID tag's prefix matches
the query sent by the reader (hence the term “query tree”), then the ID tag will transmit.
There is, of course, a possibility that more than one ID tag will have the prefix values
which match for the first m bits. In that case, the reader will detect the collision, then
transmit a longer query, until the prefix matches only one of the ID tags. The reader can
store information about which queries have already been tried and if they resulted in a
single response, a collision, or no response. Using this information, the reader can be
programmed to smartly traverse the tree of possible ID tags [6].
2.3.2.2.1.3 Binary Search

This protocol uses a similar mechanism as the query-transmission used in the “query tree” protocol, but the reader will transmit a query that has the same number of bits as the length of an ID tag's bit-sequence, and the ID tags will only respond if their ID is equal to or less than the transmitted query. In this way, the reader can always split the number of potential ID tags that can respond in half. If no collision occurs, then the reader can safely presume that all of the ID tags that have the value of the query or less have been identified, and choose its next query to be a higher ID value [6].

If Manchester encoding is used instead of an NRZ waveform, and the replies from any responding ID tags are synchronized at the bit-level, then it is possible to identify precisely which bits transmitted by the ID tags match and which do not. Then the reader can make smarter decisions about subsequent queries to send based on which bits were collided and which were not [6]. Note that the bit-by-bit synchronization is an important feature of the proposed algorithm as well.

2.3.2.2.1.4 Bitwise Arbitration

This protocol works by systematically working through all of the ID bits, from the MSB to the LSB, of all the IDs in the read-field. The reader will send a query requesting that all of the ID tags respond with their \( N^{th} \) bit. If there is no collision, then the reader can assign the received bit value to all of the ID tags in the field. If there is a collision, then the ID tag sends a command to silence all of the ID tags that responded with a particular value (for example, all the ID tags that responded with a “1” will be silenced until all of the ID tags that responded with a value of “0” are positively identified). Note
that the requirement that the reader must identify which bits were collided and which were not means that the system must use something like Manchester encoding to signal the bits, as well as bit-by-bit synchronization [6].

### 2.3.2.2 Criticism

Tree-based protocols try to attack the collision-management problem by requiring the reader to take a more active role in controlling and communicating with the ID tags, but they suffer from the same drawback as the ALOHA protocols – that is, they must experience a collision before the collisions are managed. This will necessarily lead to wasted time-slots and re-transmissions, which causes a reduction in throughput. The goal of the proposed algorithms is to construct the RFID system in such a manner that no time-slots are wasted, so that an ID tag is almost always in the process of being positively identified.

### 2.3.2.3 Other Notable Protocols

There are many other notable protocols that are too numerous to describe completely. Some of these protocols have very high performance levels, such as CSMA (which requires ID tags that can sense the channel) and its variants, which achieves 90% throughput in some conditions [9]. However, adding descriptions of these protocols would not add to the understanding of the proposed algorithms or illuminate the limitations of existing ones any further. The most important thing that must be understood about any algorithm is its performance level (which here is considered to be the throughput).
The previous protocol families (ALOHA and Tree-Based) are considered worth mentioning for two reasons: 1) They are in widespread use, and therefore are fundamental to any paper regarding RFID operation, and 2) they have features which will be used in the proposed algorithm, and are therefore relevant for their operational details as well as validation that the functional blocks of the proposed algorithm are feasible in practice.

### 2.3.2.4 Performance of RFID Anti-collision Protocols

The performance of RFID anti-collision protocols is measured in terms of the system's throughput. The literature defines throughput as the “ratio of the expected values of successful time-slots / total number of time-slots” [10]. This can also be regarded as a measure of system information efficiency, in the sense that the information out from a reader which has identified N tags is equal to N times the number of bits per tag, and the information in, or information used, by the reader in order to identify those tags is the number of time slots required to read those tags times the number of bits per time slot (which is the same as the number of bits per tag). This definition is consistent with the literature, both in conceptual terms and also with regard to the results obtained (see “Performance Analysis of Proposed Algorithm”, in which the DFSA algorithm is used as a baseline comparison to the proposed algorithm. The performance result achieved in the simulation agrees with the known results published in the literature).

The literature includes well-known throughput performance metrics for the established protocols, which are listed below [6].

1. Pure ALOHA: 18.4 %
2. Slotted ALOHA: 36.8%
3. Dynamic Framed Slotted ALOHA: 42.6%
4. Tree-Based Protocols: 43%
Chapter 3: Proposed Algorithm

This chapter will describe the proposed algorithm in sufficient detail that it can be simulated, implemented, and analyzed. The major steps of the protocol are presented first, then a set of definitions useful to the detailed discussion of the proposal are presented, followed by a detailed exposition of each step in the protocol. This chapter concludes with a detailed example.

The proposed protocol works by grouping the ID tags into subsets called “clusters”. A cluster is made up of multiple ID tags which have enough similar bits in their IDs that they can transmit their information at the same time without interfering with each other's transmission in a destructive manner. This guarantees that the ID tags will be able to transmit without collisions.

Any given population of ID tags may contain multiple clusters – but only one cluster can transmit at a time. One cluster must be selected, and this is handled through an arbitration process which establishes one of the ID tags as the most-dominant tag. The cluster that begins with the most dominant ID tag will then transmit while the others wait. Once the ID tags within a given cluster have been detected by the reader, they fall silent for the remainder of the execution of the protocol in order to allow other clusters to be identified.

This segregation of the ID tags and the order in which they are identified is shown pictorially in Illustration 3.1. The positioning of the ID tags in the illustration is meant to indicate their groupings as clusters, not any physical placement related to their spatial distribution.
The numbered tags represent the dominant ID tags which will be selected to start the transmission of a cluster of ID tags (the remaining ID tags, which are in the same cluster are not numbered). The numbers represent the order that the clusters will be identified in. In this diagram, the lower numbers represent the more-dominant ID tags, and therefore will be selected to transmit first.
3.1 Protocol Description

The protocol consists of a sequence flow that involves the steps shown below in the flow diagram in Illustration 3.2. Each step will be explained in more detail in subsequent sections, but the list below the diagram provides a brief explanation of the purpose of each step.

Flow Diagram Steps:

- **START**: This state in the sequence represents the commencement of a read-sequence, which is initiated through the user or system controlling the reader.
- **System Initialization**: This state in the sequence will establish the carrier wave read-field, and therefore provide power to the tags. The tags will initialize their internal state and then wait for a synchronization pulse from the reader.
- **Synchronization**: This state will send a pulse to all the ID tags, synchronizing them for transmission purposes.

**Illustration 3.2: Coherent Collision RFID Protocol**

- **Empty Arbitration?**
  - Yes
    - If no ID tags participated in Arbitration, then the tag identification process is complete.
  - No
    - Cluster Transmit
      - Subsequent ID tags that are part of the same cluster as the First ID will transmit.
      - A tag transmits when it detects a match for its prefix.
• First ID Arbitration
All tags which have not yet been identified will participate in an arbitration process which will establish a “most-dominant” ID tag. At the end of the arbitration period, the dominant tag will be positively identified. If no tags participate in the arbitration process, then the reader determines that all ID tags in the read-field have been identified and exits the protocol.

• Cluster Transmit
ID tags which have not yet been identified and are “next” in the ID tag sequence (which is defined more precisely later) will transmit. Depending on the specific ID tags that are present in the field, there will be “clusters” of ID tags that will all be identified at a very rapid rate before ending the cluster transmission. Clusters can be of any size, depending only on the distribution of ID tag values in the ID tag population. Each identified tag will receive confirmation from the reader that it was identified properly, and will thus enter a passive state where it does not transmit and will not participate in the next round of First-ID Arbitration.

• Return to Synchronization:
When the cluster transmission has ceased, the reader will return to the synchronization step. The ID tags which have not yet been identified (because they were not part of the “cluster” that transmitted) will respond to the synchronization pulse and participate in arbitration, beginning the process over again.

3.2 Terminology

This section defines a set of terms that are specific to the discussion of the protocol:

• ID Tag Bit-Sequence (IDTBS)
This is the bit-sequence that identifies a given ID tag. It is $L$ bits long, and no two ID tags can share the same ID Tag Bit-sequence.

• ID Tag Sequence Code (IDTSC)
This is a sequence of bits, which has length $2^L - 1$, in which every bit defines the beginning of an $L$ bit codeword, and every codeword is unique. This defines the set of all ID Tag Bit-Sequences, and also defines an ordering for the IDTBSs, since each codeword exists in only one location in the ID Tag Sequence Code. Note that the IDTSC should be considered to be a circular sequence, with no defined start or end. Otherwise, the last $L - 1$ codewords would not have sufficient bits to form complete codewords.

• ID Cluster (IDC)
An ID Cluster consists of a sequence of ID Tag Bit-Sequences which, when ordered according to the ID Tag Sequence Code, allows each ID Tag Bit-Sequence to share the last $k$ bits of the ID Tag Bit-Sequence with the first $k$ bits of the following ID Tag Bit-Sequence. The value of $k$ can be different from
one *ID Tag Bit-Sequence* overlap to the next, but must always equal at least one in order to allow the cluster to continue. The concept of an ID Cluster is shown in Illustration 3.3.

- **Dominant Bit (DB)**
  A Dominant Bit is a single bit of information transmitted by an ID tag which causes the reader to register a drop in the voltage across its antenna terminals. In the type of system under consideration, this means that the ID tag is engaging its loading circuit.

- **Recessive Bit (RB)**
  A Recessive Bit is a single bit of information transmitted by an ID tag which is not dominant – meaning that it does not cause a voltage drop across the reader’s antenna. In the type of system under consideration, this means that the ID tag is *not* engaging its loading circuit.

**3.3 Definition of ID Tag States**

The following state diagram in Illustration 3.4 shows the possible states that an ID tag can assume. The diagram is followed by a detailed definition of each state. Note that all states have an implicit transition to the “OFF” state, which can occur if the ID tag...
cannot power itself from the carrier wave field anymore. Those transitions are not shown in the interest of clarity.

Illustration 3.4: Coherent Collision ID Tag State Diagram

- **INIT:**
  An ID tag in this state is waiting for a synchronization pulse from the reader to allow it to enter the arbitration state.

- **ARBITRATION:**
  An ID tag in this state is participating in the First-ID arbitration process.

- **WAITING:**
  An ID tag in this state has completed arbitration, but was not selected as the most dominant ID tag. It is now waiting for the bits received by the reader to match its prefix. Additionally, if an error occurs in any of the other states, the WAITING state allows the ID tag to try again when the next arbitration happens.

- **TRANSMITTING:**
  An ID tag in this state is transmitting its IDTBS.

- **ENDING:**
  An ID tag in this state is waiting for the reader to send an acknowledgement that it has been identified. This ensures that the ID tag will not enter the PASSIVE state before being positively identified.
• PASSIVE:
  An ID tag in this state has been positively identified by the reader, and will not
  transmit any more information until after the next time that it loses power.

3.4 System Initialization

This step is initiated by some external actor, such as a human operator who presses a
button on the reader to start scanning, or a software program that sends a command to
the reader to initiate the reading sequence.

The reader will initialize its internal state to prepare to scan its read-field for ID tags.
The reader will also establish the carrier field by energizing its transmitting antenna with
an oscillating current at the default carrier wave frequency, which we will call $f_0$. The ID
tags all have antennas which have a resonance at frequency $f_0$, allowing them to
efficiently use the power provided by the field.

The ID tags are responsible for turning on and initializing their internal state within a
certain maximum time window. Then the ID tags will wait in the INIT state until they
detect a synchronization pulse from the reader.

3.5 Synchronization Pulse

The reader will emit a synchronization pulse after the system initialization phase has
completed. The exact nature of the pulse is unimportant to the design of the protocol.
All the ID tags will use the synchronization pulse as a signal to enter the arbitration
state, as well as synchronizing all of their internal clocks. Clock synchronization may be
retained throughout the read process by using the oscillation of the read-field as the
clock signal to the ID tags' timing circuitry.
3.6 First ID Arbitration

First-ID Arbitration is a process that will allow all of the ID tags which have yet to be positively identified to determine a “most-dominant” ID tag. This is then used as the first identified ID, and subsequent IDs will be read in the order determined by the IDTSC. When executed as proposed below, the most dominant ID tag will be the one with the highest binary value (if one considers the first transmitted bit to be the most-significant bit, i.e. big-endian).

The ID tags will compete for dominance each time a bit is transmitted. If an ID tag transmits a recessive bit, and any other ID tag transmits a dominant bit at the same time, then the ID tag transmitting the recessive bit will lose the arbitration and must go silent for the remainder of the arbitration process. In order to implement this, there must be a mechanism by which the ID tag is immediately informed of the result of the each transmitted bit. There is a short time-slot inserted after the transmission of each bit to allow the reader to transmit this information back to the ID tags before the commencement of the next bit transmission. See the section on “Reader-Feedback Mechanism” below for details.

The concept is illustrated in Illustration 3.5. This example shows three 8-bit IDTBS ID tags competing for dominance, with the blue boxes indicating dominant bit transmissions and the red indicating recessive bit transmissions. The black diamond over a recessive bit indicates that the ID tag lost arbitration at that point – and all subsequent bits in that ID tag’s IDTBS will not be transmitted. The bits received by the reader are shown at the bottom of the figure.
Notice that the received bits match precisely the IDTBS claimed by Tag 3, and that Tag 3 is the ID which wins the arbitration. This will always be the case, so, at the end of arbitration, the most dominant ID tag will be identified by the reader and will also be known to all participating ID tags because of the bit-by-bit feedback provided by the reader.

All of the ID tags which participate in arbitration but are not dominant will shift into the WAITING state. If no ID tags participate in arbitration, then the reader presumes that all ID tags that are present in the read-field are in the PASSIVE state, and therefore have already been identified. The reader can then exit the protocol and report its resulting list of detected ID tags.

### 3.7 Cluster Transmit

The remaining ID tags which participated in arbitration but were not the dominant tag will have shifted into the WAITING state by the end of arbitration. The purpose of the Cluster Transmit portion of the protocol is to allow all of the ID tags which are part of the same cluster as the dominant ID tag to transmit their IDTBSs with a guarantee of being identified without collision (barring any communication errors).
The ID tags must all contain a prefix-matching function (similar to what is used in the “Query Tree” protocol covered in the “Tree-Based” section above) which monitors the bits received by the reader through the reader-feedback mechanism and stores them in a “Reader Feedback Register” (RFR). The RFR has length L bits. The prefix-matching circuit is looking for an L-bit sequence which matches the value stored in its prefix-matching register. The prefix-matching register contains the L bits prior to the ID tag's IDTBS in the IDTSC. This allows the ID tag to be aware when the reader has received the L bits that precede the ID tag's IDTBS in the IDTSC. When this occurs, the ID tag will shift into the TRANSMITTING state, causing the ID tag to transmit its ID to the reader. The concept is shown in Illustration 3.6.

**Illustration 3.6: Prefix Matching**

Note that during the first-ID arbitration sequence, the subsequent ID tags which should start transmitting coherently during the dominant ID tag's transmission will not be able to start transmission, because they will not have received sufficient bits to fill up the L-bit long RFR. Therefore, there must be some other means that the ID tag uses during the
first L bits after the start of arbitration to ensure that the bit-sequence received by the reader is not interrupted.

This is accomplished by allowing the ID tag to exit the WAITING state and move to the TRANSMITTING state whenever any L-bit subset of the sequence formed by the PMR plus the IDTBS is matched. When this occurs, the ID tag will start transmitting its IDTBS at the bit immediately following the last bit in the RFR. This will allow the reader to receive the next bit in the IDTSC without interruption. This situation is shown in Illustration 3.7.

Each ID tag that has engaged in arbitration but which did not “win” will be waiting for a prefix-match to allow it to transmit. Therefore, as long as there is even a single ID tag which overlaps with the currently transmitting ID tag, the reader will continue to receive bits from the ID tags. In this manner, an entire string of ID tags which are all part of a single cluster in the IDTSC will transmit without collision. Hence the name “Cluster Transmit”. Each ID tag, when done transmitting, will transition to the PASSIVE state.
after receiving acknowledgement that it has been positively identified from the reader (see “ID Tag Identification” below).

### 3.8 Return to Synchronization

The Cluster Transmit sequence will end when the reader identifies that the IDTSC has been violated. This will occur because the end of the cluster has been reached, and the last ID tag has transmitted its last bit, and no other ID tag has matched its prefix and therefore is not transmitting. The reader will expect to see a certain sequence of dominant and recessive bits, according to the IDTSC. At the moment the reader expects a dominant bit, it will not receive it because no ID tag is transmitting. The reader will then re-issue a synchronization pulse, and all of the ID tags that are still in the WAITING state will return to the ARBITRATION state. This allows the system to quickly move to the next cluster which contains the next most-dominant ID tag.

As stated in the section “First ID Arbitration”, if no ID tags participate in arbitration, then the reader will exit the protocol and the reading sequence is complete.

### 3.9 Reader-Feedback Mechanism

The previous sections described the proposed protocol completely, but the protocol does require another underlying mechanism which must be defined. There must be some way that the reader can transmit the value of the bits received to the ID tags so that each ID tag can add the bit to its RFR for prefix-matching purposes, and also so that the ID tag can confirm that its transmission was received correctly. This is accomplished by inserting a short time period during each bit transmission window in
which the reader will modulate the read-field carrier wave in a predefined manner to indicate to the ID tags what type of bit was received – dominant or recessive. The precise form of the modulation is not important to the protocol. One example is that the reader might change the frequency of the carrier wave, perhaps to a harmonic or sub-harmonic of the base carrier wave frequency, $f_0$. The only requirements are that the ID tags still have sufficient resonance in their receiving antennas to power their internal circuitry, and that the ID tags are capable of detecting the difference in the signals. The proposal is shown in Illustration 3.8.

![Illustration 3.8: Bit-Frame Timing Diagram](image)

3.10 ID Tag Identification

The protocol, as discussed thus far, will allow the reader and ID tags to function coherently in order to reconstruct the portion of the IDTSC which contains all of the ID tags. However, it is still not completely possible for the reader to determine exactly which ID tags are present in the read-field, because any combination of IDTBS that are subsets of the received bit-sequence will technically be valid choices. The problem is depicted in Illustration 3.9.
This ambiguity is removed by requiring that all ID tags transmit an “End of Transmission” (EOT) flag at the end of successfully transmitting their entire IDTBS. The ID tag will not know if the $L^{th}$ and final bit of the IDTBS was received without corruption at the reader until the end of the $L^{th}$ bit-frame of transmission, so the earliest opportunity to transmit the EOT signal would be during the $(L+1)^{th}$ bit-frame. The EOT transmission must not interfere with other ID tags’ transmissions, so it must not occur during the first half of the bit-frame. Instead, the EOT transmission (which consists of a single dominant pulse) occurs during the second half, when the other ID tags are receiving feedback from the reader. The reader will take this extra dominant bit as an indication that the previously received $L$ bits are the IDTBS of an ID tag which is present in the read-field. The reader then sends acknowledgment during the first half of the next $(L+2)^{th}$ bit-frame, while the other ID tags are transmitting. This method ensures that the ID tag does not enter the passive state without first confirming that the reader has positively identified the tag. The EOT communication sequence is shown in Illustration 3.10.
3.11 Example

The operation of the algorithm may become more clear for the reader by presenting an example. Consider a situation where a system has ID tags that have 5 bits in each ID. The IDTSC will have length $2^L - 1 = 31$ bits, and there are up to 31 unique ID tags that can be identified. The IDTSC arbitrarily selected for this example is:

\{0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1\}

Consider that the system has ten ID tags present in the read-field. The randomly selected values of the ID tags (these numbers represent the IDTBSs as decimal...
numbers) in the read field are:

\{1, 4, 10, 12, 19, 22, 23, 24, 26, 28\}

The progression of the system is shown in Illustration 3.11. Each successive row corresponds to an incremental time-step equal to a single bit-time – i.e. enough time for the ID tags to transmit a single bit to the reader, and the reader to respond with its reader-feedback. Each line displays the state and transmitted/received bits of the ID tags and reader at the end of each time step. The reader's received-bit on each line represents the logical OR of all of the ID tags' outputs on the same line, and the reader's state is updated internally before the end of the time-step. Therefore, for reader-driven state transitions (i.e. the beginning of arbitration), the reader will first show that its state has changed, and the ID tags will respond appropriately in the next line.

The state and load-modulation output of the ID tags are shown in the columns labeled “State” and “Output”, respectively, for each ID tag. The states are coded by the first letter of the state name shown in Illustration 3.4: Coherent Collision ID Tag State Diagram. Note that there is an additional “S” state, which corresponds to the “Receive Sync Pulse” transitions in the diagram. The dominant and recessive bits are represented by a “1” and “0”, respectively.

The state of the reader and its received-bit is shown on the right side of the “X” out column. The states of the reader are similarly identified according to the states shown in Illustration 3.2: Coherent Collision RFID Protocol (except “SYSTEM INIT”, which is represented by “I”, and “FIRST ID ARBITRATION”, which is represented by “A”). The expected bit, which is only known after the completion of an arbitration sequence, is
shown in the column “Expt’d Bit”. The bits shown in square brackets [ ] represent the bits received during arbitration – the reader does not have any way to anticipate the value of the next received-bit until after arbitration. If the received-bit does not match the expected bit, then the received-bit is colored red.

The tags that have been successfully identified by the reader are shown in the next column.

The “State” columns are highlighted green whenever a state transition occurs, and the “PASSIVE” state is highlighted red, to aid reading the table.

3.11.1 Example Narration

The system begins with all of the tags in the OFF state, and the reader starts the read process in the INIT state. Note that the amount of time spent in the INIT and SYNC states will depend on specific factors regarding the design of the tags and readers which are beyond the scope of the proposed protocol. Here they are considered to be a single time-step.

The ID tags go through their initialization sequence, and the reader sends the start of a SYNC pulse (represented here by a “SYNC” state). The ID tags respond by entering the SYNC state. At this point, all of the ID tags and the reader are synchronized. When the reader concludes the SYNC pulse, it enters the ARIBTRATION state, and the ID tags respond in the next time-step by also entering the ARBITRATION state. In the first round of arbitration, ID tags #1, #19, and #23 all transmit a “1” - the other ID tags transmit a “0”. Therefore, by the next time-step, all other ID tags are in the “WAITING” state. Tag #23 finally wins the first arbitration, and identifies itself to the reader by
sending the “EOT” pulse (in the “E”, or “ENDING”, state), before finally entering the PASSIVE state. Note that ID tag #10 begins transmitting immediately after Tag #23 stops, in time-step 9, and transmits for only 3 bits before ending and being identified. This is because ID tag #10 reached a prefix-match at the end of arbitration which required it to start transmitting from its third bit in order to ensure that the reader received the proper sequence of bits to continue the portion of the IDTSC started by ID tag #23 (see Illustration 3.7: Prefix Matching - During Arbitration). The reception of an additional bit beyond the end of arbitration which matches the expected bit according to the IDTSC allows the reader to enter the “CLUSTER TRANSMIT” state. The reader will remain in that state until the received-bit sequence does not match the IDTSC.

The reader continues to receive a bit sequence that matches the IDTSC because ID tags 1, 4, 26, 19 and 28 are each able to match their prefix, and therefore begin transmitting their IDTBS. These ID tags constitute the first cluster that is identified in this example. The cluster ends in time-step 27 (but the reader remains in CLUSTER TRANSMIT until the end of the next time-step, in order to give an opportunity to process the EOT pulse from ID tag #28 – see section “ID Tag Identification”), when the reader received a “0” but expected a “1”. The reader then enters the SYNC state, starting the arbitration process again.

The second arbitration process only includes ID tags #12, #22, and #24. ID tag #22 wins, but no other tag is able to obtain a prefix match. The cluster ends immediately after arbitration, with a “0” received when expecting a “1”. The third arbitration is won by ID tag #12, and ends immediately after arbitration also. Finally, the fourth arbitration is won by ID tag #24. This fourth cluster contains only one tag, just like the last two
clusters, but it continues for an additional bit beyond arbitration because the subsequent bit in the IDTSC that is expected immediately after arbitration is a “0”, which is the received-bit even when no ID tags are transmitting. It is not until the next bit, when a “1” is expected but a “0” received, that the reader is able to determine that the cluster has ended and therefore force another arbitration to take place. This final arbitration is empty – that is, no ID tags participate – because all of the tags have been identified and are in the PASSIVE state. The reader ends the reading process at time-step 60. The total amount of bits contained in the identified tags is $10 \times 5 = 50$ bits. The efficiency is

$$\frac{50}{60} = 0.83 \text{, or 83\%}.$$
Chapter 4: Performance Analysis of Proposed Algorithm

This chapter will cover the simulation methodology used and results obtained in evaluating the performance of the proposed protocol. Results are also computed for a known baseline algorithm in order to serve as a comparison and also to confirm that the simulation can produce results that agree with existing literature.

4.1 Experimental Methodology

The experiment to measure the algorithm's performance was constructed to test the algorithm over a variety of test conditions. The independent variables are listed in Table 1, along with the range that they assumed during the experiment. The dependent variables are also shown in the table.

The central metric of interest is the throughput, which, as discussed above in the section “Performance of RFID Anti-collision Protocols”, is a measure of the ratio of the quantity of detected information to the amount of transmitted and received information required in order to detect the information.
Table 1: Experimental Independent and Dependent Variables

<table>
<thead>
<tr>
<th>Independent Variables</th>
<th>Tested Range</th>
<th>Dependent Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L</strong></td>
<td>[8, 10, 12]</td>
<td><strong>T</strong></td>
<td>Time required to successfully read all of the tags. Measured in number of bit-transmissions required, or bit-time.</td>
</tr>
<tr>
<td>Number of bits per IDTBS.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>N</strong></td>
<td>[1, 4, 8, 12, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]^2</td>
<td><strong>TP</strong></td>
<td>Throughput</td>
</tr>
<tr>
<td>Number of ID tags in the read-field.</td>
<td></td>
<td></td>
<td>$TP = \frac{N \times L}{T}$</td>
</tr>
</tbody>
</table>

The specific ID tags used in each simulation were randomly selected from the population of all possible tags, with no duplicates. In order to gain an understanding of the expected value of each of the dependent variables, the simulation was run 32 times under each test condition, and the results averaged. Each time the simulation was run again, a new set of random ID tags was selected. In the case of Coherent Collision, a new IDTSC was also randomly generated each time the simulation was run.

---

2 In cases where the number of ID tags is equal to $2^L$, the number of tags was reduced by one because the maximum number of tags in the system is $2^L - 1$. If the number of ID tags exceeded the maximum by anything more than one tag, then the condition was not tested, since it would require duplicate tags.
4.2 Simulation Methodology

The author developed a model of the RFID system using object-oriented programming techniques in order to create a realistic simulation. The object-oriented framework provided a good means to ensure that the reader and ID tags only communicated in the same way that would occur in the real world - that is, through load-modulation (ID tags to reader) and carrier-wave modulation (reader to ID tags). The objects created were as follows:

1. **RFID System** – The RFID System object is capable of creating and destroying ID tags and a reader, and is also capable of tracking the status of the objects in the system. The system is also responsible for defining the duration of the INIT and SYNC states, the number of bits per ID (“L”), the number of ID tags in the system (“N”), the allowable carrier wave frequencies and their associated state, and also determines the IDTSC (for Coherent Collision) or the tag-estimation function and associated parameters (for DFSA). These parameters were communicated, as appropriate, to the reader or ID tags when they were created.

2. **RFID Reader** – The RFID Reader object is capable of executing the protocol under test (Coherent Collision or DFSA). It keeps track of the identified ID tags, provides a means for initiating a read (i.e. a “Start Read” button), generates a carrier wave, and reads the load-modulation feedback from the tags. The reader was modeled as an object with a carrier-wave frequency output and a received-bit input. The carrier-wave frequency output value represented the driving frequency that the reader would create in its output antenna. The received-bit input represented the reader's detection of the ID tags' binary transmissions to the reader via load-modulation.

3. **ID Tag** – The ID Tag object is also capable of executing the protocol under test. It keeps track of its own state, knows its own IDTBS, and has the necessary internal counters and registers needed to execute whichever protocol it is designed for. The ID tags were modeled as an object with a carrier-wave frequency input and a transmitted-bit output. The carrier-wave frequency input represented the frequency of the carrier-wave impinging on the ID tag's antenna, and the transmitted-bit output represented the state of the ID tag's load-modulation circuitry.

The reader-to-tag communication (and power) mechanism was simulated by copying the reader's carrier-wave frequency value output to each of the ID tags' carrier-wave input value (regardless of the ID tag's state). The tag-to-reader communication via load-
modulation was simulated by OR’ing together all of the transmitted-bit output values of each of the ID tags (again, regardless of state) and writing the resulting value to the received-bit input of the reader.

Each run of the simulation was considered complete when the reader reached the exit condition of its algorithm. Note that merely identifying all of the ID tags was not sufficient – the reader must also determine that there are no more ID tags left to be identified.

4.3 Model Parameters and Assumptions

The model uses a simulation bit-time penalty of six bit-times for each sync pulse. The simulation also assumes the following are true:

- All bits are successfully transmitted and received between tags and readers.
- Bit-level synchronization is maintained properly between each tag and the reader.
- The population of ID tags is static during the read process.
- There are no ID tags in the PASSIVE state at the start of the reading process.

4.4 Baseline Comparison

The Dynamic Framed Slotted ALOHA algorithm was used as a baseline comparison to the proposed algorithm. The default frame size was initialized at 32 frames, the Vogt-I function (see [6]) was used as the tag-estimation function, and the frame size was unlimited and restricted only to be a positive integer.
4.5 Comparative Bounds and Strategies

In this section, the author will present a couple of simple alternative strategies and an upper bound that can be used as a comparison to the efficiency of the proposed algorithm. The two strategies represent stripped-down versions of Coherent Collision which correspond to the operation of the algorithm at very low or very high tag densities. The upper bound represents the expected minimal amount of information required in order to identify a random population of ID Tags.

4.5.1 Repetitive Dominant Tag Strategy

A strategy can be constructed which consists only of repetitively identifying the dominant ID tag, without executing the cluster transmit portion of the Coherent Collision algorithm. This is very similar to how the proposed algorithm works when the ID tag population has a low-density and is likely to be highly fragmented (i.e. most ID tags are not part of a larger cluster). The most significant difference is that in Coherent Collision, the algorithm suffers a penalty for each cluster because it must determine that the cluster has ended by checking for a violation of the IDTSC – this strategy can forgo that check because it does not execute a cluster transmit but instead returns directly to the beginning of arbitration as soon as the dominant ID tag has been identified. For each tag identified, the time required will be the sum of the sync pulse duration (S), an arbitration period with length equal to the number of bits per IDTBS (L), plus a single bit-time to instruct the ID tags to enter arbitration (as the reader can verify by close inspection of the amount of time that the RFID reader spends in arbitration in the example shown in Illustration 3.11), plus an additional bit time for the reader to receive
the EOT bit. Finally, the reader must conclude that there are no more tags to read by executing an empty arbitration at the end of its sequence. Thus the efficiency of this strategy (which will be termed “Repetitive Dominant Tag”, or RDT) is given in Equation (4.1).

\[
E_{\text{RDT}}(N) = \frac{N \times L}{(N+1)(S+L+2)}
\]  (4.1)

4.5.2 Complete Code Sequence Strategy

Alternatively, another simple strategy would be to allow the reader to choose an arbitrary starting point in the IDTSC, then send the IDTSC bits sequentially through the reader-feedback mechanism to the ID tags, and allow each ID tag to transmit an EOT pulse to announce its presence when the ID tag senses a match between the reader-transmitted bits and its IDTBS. This is very similar to how the Coherent Collision algorithm behaves when the tag densities are very high. This would require that the reader run through \(2^L - 1\) iterations, plus an additional \(L - 1\) iterations in order cover all the possible IDTBS values. Thus the efficiency of this strategy (which will be termed “Complete Code Sequence”, or CCS) is given in Equation (4.2).

\[
E_{\text{CCS}}(N) = \frac{N \times L}{2^L + L - 2}
\]  (4.2)

4.5.3 Minimum Information Bound

An upper bound on the performance of the algorithm can be calculated by considering the expected minimum amount of information that must be transmitted to the reader in order to identify the tags. Given N tags with L bits per tag, the algorithm must, at a
minimum, visit every bit in the IDTSC which is part of at least one IDTBS of the ID tags. This represents the minimum information that must be transmitted from the ID tag population to the reader. A lower bound on the expected execution time of the algorithm, then, is the expected number of bits which are part of at least one IDTBS. This is the same as the total number of bits in the IDTSC multiplied by the probability that any given bit is part of at least one IDTBS.

Consider the probability that a given bit in the IDTSC is \( \text{not} \) part of an arbitrarily chosen IDTBS. That probability is equal to the number of IDTBSs which \( \text{do not} \) contain the given bit divided by the total number of IDTBSs. All of the available IDTBSs do not contain the given bit except for \( L \) of them. Therefore, the probability that a given bit in the IDTSC is not part of any arbitrarily chosen “first” IDTBS is \( \frac{2^L - 1 - L}{2^L - 1} \).

If the bit in question is not part of the “first” IDTBS, consider the conditional probability that it is then not part of another (arbitrarily chosen) “second” IDTBS. The total number of IDTBSs which are available to be chosen as the “second” IDTBS is reduced by one from the number available to be chosen as the first. However, the number of IDTBSs which contain the bit is still \( L \). Therefore, given that a random bit in the IDTCS is not part of the “first” IDTBS, the conditional probability of it not being the “second” IDTBS is \( \frac{2^L - 2 - L}{2^L - 2} \).

The same argument applies to a third, fourth, and fifth IDTBS, and so on through the entire population of ID tags. This therefore allows us to write the conditional probability
that a given bit “b” in the IDTSC is not part of the \( n \)th IDTBS, given that it is not part of any of the previous \([1..n-1]\) IDTBSs, as shown in Equation (4.3):

\[
P(b \notin IDTBS_n \mid b \notin IDTBS_{0..n-1}) = \left( \frac{2^L - n - L}{2^L - n} \right) \tag{4.3}
\]

The probability that the given bit is not part of any of the IDTBSs is then the product of all of the conditional probabilities, as shown in Equation (4.4).

\[
P(b \notin IDTBS_n \forall n \in [1..N]) = \prod_{n=1}^{N} \left( \frac{2^L - n - L}{2^L - n} \right) \tag{4.4}
\]

Given (4.4), it can then be stated that the probability that a given bit is part of an IDTBS, and therefore one of the bits that must be “visited” by the minimum information algorithm, is the complement of the probability provided in (4.4).

\[
P(\exists n \in [1..N] : b \in IDTBS_n) = 1 - \prod_{n=1}^{N} \left( \frac{2^L - n - L}{2^L - n} \right) \tag{4.5}
\]

Therefore, the expected number of bits that are part of at least one IDTBS, and therefore the expected minimum amount of bit-time required to identify the ID tags, is equal to Equation (4.5) multiplied by the total number of bits in the IDTSC:

\[
E_{min\text{info\_bits}}(N) = (2^L - 1)^* \left[ 1 - \prod_{n=1}^{N} \left( \frac{2^L - n - L}{2^L - n} \right) \right] \tag{4.6}
\]

The average efficiency, then, cannot be greater than \( N*L \) divided by Equation (4.6).

This expression, shown in Equation (4.7), serves as an upper bound on the expected performance of the Coherent Collision algorithm.
\[ E_{\text{mininfo}}(N) = \frac{N \cdot L}{(2^L - 1) \times \left(1 - \prod_{n=1}^{N} \left(\frac{2^L - n - L}{2^L - n}\right)\right)} \] (4.7)

Note that this is not a strict upper bound – there are some contrived cases where much higher performance could be experienced, such as when all of the ID tags are part of a single cluster which starts with the dominant ID tag. Rather, this equation represents the upper bound on the expected, or average, performance of this type of algorithm.

### 4.6 Results

The results are shown in Illustrations 4.1 through 4.6. The performance at lower field densities (less than approximately 20%) is shown on the following page from the graph which shows the complete range of field densities.

The performance of the DFSA algorithm is dwarfed by the efficiency of Coherent Collision, especially at high field densities. As one might expect, the overhead inherent to both algorithms means that lower efficiency is experienced at very low field densities (on the order of a handful of tags), but even there Coherent Collision outperforms DFSA.

The dotted green line represents the performance of a RDT system, and the dashed-dotted maroon line represents the performance of a CCS system. The CCS algorithm does better at high densities and the RDT algorithm does better at low densities.

Any general-purpose anti-collision algorithm must, of course, be able to perform at any field density. Therefore, the Coherent Collision pays a small price for its flexibility compared to the best of either RDT or CCS at nearly all field densities. Fortunately, it
appears that Coherent Collision does a fairly good job of staying close to the higher performing comparative bound between RDT and CCS. At low densities, the efficiency is close to RDT.

As the field saturation increases to about 3%, the Coherent Collision algorithm begins to take advantage of the efficiency that comes from clustering ID tags together. This can be clearly seen in the graphs by observing where the Coherent Collision algorithm’s efficiency begins to exceed the RDT bound, which can only happen when the cluster transmit portion of the algorithm is active. The efficiency approaches that of the CCS method as the field saturation approaches 100%. Another way to view this is that the algorithm is good at choosing to operate as either an RDT-like algorithm or as a CCS-like algorithm.

The minimum information bound expressed in Equation (4.7) is shown using a long-dashed light blue line. The CCS bound and the Coherent Collision algorithm performance all approach (but always remain slightly below) the minimum information bound.
Illustration 4.1: Algorithm Efficiency - ID Length = 8 Bits
Illustration 4.2: Algorithm Efficiency (Low Density Region) - ID Length = 8 Bits
Algorithm Efficiency

Coherent Collision vs. DFSA (10-bit)

Illustration 4.3: Algorithm Efficiency - ID Length = 10 Bits
Algorithm Efficiency

Coherent Collision vs. DFSA (10-bit)
Low-Density Region

Illustration 4.4: Algorithm Efficiency (Low Density Region) - ID Length = 10 Bits
Algorithm Efficiency

Coherent Collision vs. DFSA (12-bit)

Illustration 4.5: Algorithm Efficiency - ID Length = 12 Bits
Illustration 4.6: Algorithm Efficiency (Low Density Region) - ID Length = 12 Bits
Chapter 5: Conclusions

The results and analysis are sufficient to conclude that the proposed algorithm has superior performance compared to the baseline DFSA algorithm.

Future research could focus on finding performance differences that may be incurred depending on the construction of the IDTSC, ways to simplify the algorithm, and finding tighter performance bounds that are based on statistical expectations of the distribution of IDTBSs in the read fields. Additionally, it would be beneficial to find a way to eliminate the need to segregate each bit-time to contain separate time quanta for ID-tag transmission and reader feedback. Perhaps novel constructions of the IDTSC could accomplish that.

The algorithm approaches the minimum information bound at high field densities, but there is clear room to improve at lower field densities. It would be interesting to find improvements to the algorithm which increase its performance near the lower end of field densities.

Additionally, the results above seem to imply a fairly consistent break-off point where the algorithm's performance begins to do better than the lower bound represented by RDT at approximately 3% field saturation. The results have only been calculated for 8, 10, and 12 bit length IDTBSs, so it would be interesting to see if the break-off point remains consistent with higher IDTBS length or if it perhaps follows a predictable trend. A more precise definition and measurement of the break-off point would also be interesting. If the trend is in fact consistent and predictable, it would be interesting to understand the underlying fundamental property of the algorithm which gives rise to that trend.
An analysis of the algorithm’s robustness against various types of communication errors and potential improvements would also be an interesting direction of research.

As pointed out in footnote 1 on page 4, it is conceivable that a similar protocol inspired by the same type of approach as Coherent Collision can be used to manage multiple readers that are attempting to read an ID tag population cooperatively in the same physical locality. This could also be a direction of future research.
References


