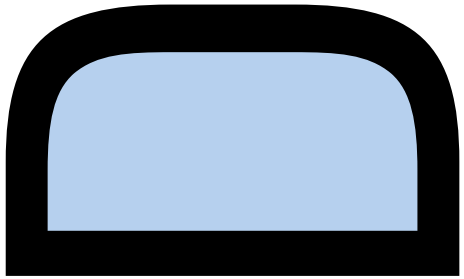
WHITE PAPER SERIES / EDITION 1



AUTO–ID LABS

BUSINESS PROCESSES & APPLICATIONS
SOFTWARE & NETWORK
HARDWARE

AUTOIDLABS-WP-SWNET-010

On Ghost Reads in RFID Systems

Daniel W. Engels

Auto-ID Laboratories
Massachusetts Institute of Technology
Cambridge, MA USA 02139

www.autoidlabs.org

Abstract

Radio Frequency Identification (RFID) systems are beginning to be used for ubiquitous object identification within the supply chain. Ubiquitous object identification enables extreme efficiencies within the supply chain. However, errors in the collected data can degrade the benefits or even increase costs as compared to not using RFID systems. The two main sources of data collection error are missed RFID tag reads and Ghost tag reads. A missed tag read results in no data, such as the identifier stored on the tag, being collected from a tag by a specific tag reader. A Ghost tag read results in erroneous data, specifically an identifier that is not stored on any tag within the reader's field, being "read" by a reader and reported as correct data. Both forms of erroneous data collection can have a significant impact on the benefits of using RFID systems; therefore, they must both be minimized. In this paper, we examine the factors that impact the appearance of Ghost tag reads and conclude that a good RFID reader design that utilizes all of the error detection features available in the RFID air-interface protocol is required to effectively eliminate the appearance and impact of Ghost reads.

1. Introduction

RADIO Frequency Identification (RFID) systems are an example of Automated Identification and Data Capture (AIDC) systems that enable a plethora of object-centric applications. The applications enabled by RFID systems promise to improve operational efficiencies, thereby reducing costs, increasing revenues, and improving safety and security. The promised efficiencies enabled by RFID systems have prompted several organizations, including the world's two largest organizations – Wal*Mart and the US Department of Defense (DoD), to begin using RFID systems for object identification within their supply chains.

The goal of using RFID systems in these organizations is to provide accurate, real-time visibility of all objects within the supply chain with minimal human intervention. Achieving this goal requires ubiquitous, or at least sufficiently ubiquitous, RFID reader deployments. A widespread RFID deployment yields detailed asset visibility by providing accurate realtime item-level location information (track capabilities) and accurate movement history (trace capabilities). When combined with an intelligent information system such as the EPC Network, accurate track and trace information enables great efficiencies in supply chain management that may be achieved with item level and RFID carried information only.

The widespread use of and reliance upon RFID systems to accurately capture data and information on the location and movement of objects within supply chains places a great importance on the validity and completeness of the data being collected. Missed tag reads, or even more nefarious, the capture of incorrect data that appears to be valid—a *Ghost tag read*, place a great burden on the information systems processing the data; therefore, these errors must be eliminated, or at least minimized and made, to the extent possible, predictable.

Missed tag reads may be minimized through many factors including good tag antenna design, radio frequency friendly product packaging designs, good RFID system installations and designs, process changes to increase tag dwell time in a reader's communication zone, and the management of reader collisions [1][2]. Missed tag reads cannot be eliminated in all operating scenarios, but they can be managed to the point that missed tag reads are predictable with some probability and small standard deviation in specific scenarios. Predictability enables the information system to be designed to manage the absence of captured data, i.e., missed tag reads, by using techniques such as identity association. The complete treatment of missed tagged reads, minimizing them and determining their likelihood, is beyond the scope of this paper.

Even more nefarious than missed tag reads are *Ghost tag reads*. A Ghost tag read, or simply Ghost read, is the reception by the reader of an apparently valid series of tag communications that is decoded into an identifier that is not stored on any tag in the reader's communication zone. A Ghost read is the receipt by the reader of data that is interpreted as valid but was not communicated by a single tag, i.e. the reader receives incorrect data which it interprets as valid data. The identifier received as a result of a Ghost read may never have been issued by the numbering authority, may be a valid identifier but not exist on any tag received at the facility experiencing that Ghost read, or may exist at that facility but is not in an area where it could possibly be read by the reader experiencing the Ghost read.

Ghost reads, in contrast to missed tag reads, are very difficult for the information system to manage. The information system must be capable of properly identifying and dispatching Ghost reads. This requires tight integration into all of the information systems, including operations and finance, used within a facility as well as accurate knowledge of what identifiers are to actually appear within a facility. Achievement of accurate, real-time information is today difficult at best and not possible at worst. Human errors, incorrect deliveries, and missing data all compound the difficulties. Consequently, the appearance of Ghost reads can have a significant detrimental effect on the operations and finances of an organization.

Fortunately, Ghost reads can be practically eliminated. The combination of correct RFID protocol design; redundant error detection features built into the air interface of the protocol, especially error detection capabilities within its logical comunication, error detection capabilities in the communication signaling, error detection in the framing of the communication, and error protection on the communicated data combine to enable, in implementations of suitably chosen protocols, the effective elimination of Ghost reads. To practically eliminate Ghost reads, it is incumbent upon the protocol designer to understand the causes of Ghost reads and make suitable selection of protocol features and on the RFID reader designer to utilize all of the available error detection capabilities. Inappropriate selection of protocol, or the use of readers that do not properly utilize all of the available error detection features available, will cause significantly increased occurrences of Ghost reads.

In the remainder of this paper, we examine the principle factors that cause Ghost reads and the protocol features that can be utilized to all but eliminate their occurrence. Error detection capabilities are built into all levels of the communication from the tag, and it is up to the reader to utilize these capabilities. We begin in Section II by examining the RFID air interface logical protocol. The logical protocol contains features that can be utilized to enhance a reader's ability to detect communication errors. We examine the error detection role of the identity discovery algorithm in

Section III, the role of communication frames, or packets, in Section IV, and the role of signaling in Section V. The data being communicated must also be protected. In Section VI we evaluate the capabilities of a checksum in detecting errors in the decoded data. A final source for the introduction of bad data into the network, through *apparent Ghost reads*, is the corruption of on-tag memory. In Section VII we examine the potential for memory errors on the tags to cause apparent Ghost reads. We briefly examine how a reader may increase its performance at the expense of increasing the probability of Ghost reads in Section VIII. We draw the relevant conclusions in Section IX.

2. RFID Protocol Considerations

An RFID air interface protocol, or simply protocol, defines all aspects of the base communication, operation, and computation capabilities of a tag and a reader. The protocol includes the definition for how a reader communicates with a tag, how a tag communicates with a reader, what information is stored on the tag, where in logical tag memory specific information is stored, and how the reader accesses the functionality and memory of the tag (both through commands, e.g., read memory, and through tag operation, e.g., reader-talks-first).

In a simplified manner, we will consider a protocol to contain three basic functional components: identity discovery, communication, and higher functionality. The identity discovery component includes the commands, algorithms, and on-tag features required to identify tags in the reader's communication zone. The communication component includes all aspects of the physical communication including symbol encoding, modulation, and timing parameters, as well as how the logical communication is packetized. Finally, the higher functionality component includes the commands, algorithms, on-tag features, and additional communication features required to perform some functionality, such as a global user programmable memory write, beyond the retrieval of a tag's stored identifier.

The Ghost reads problem is affected by the identity discovery and communication components only. The identity discovery and communication components are a tightly interconnected part of the specification that, for low-cost tags, assumes the tag has limited capabilities in its communications and computations capabilities. For passive tags, the communications will be weak compared to the reader's communication signals and potentially as compared to the noise in the environment. As a tightly connected specification, all of the protocol features contribute to the ability of a reader to detect errors in the communication from the tag to the reader. Undetected errors in the tag to reader communication are the primary source of Ghost reads.

Tag protocols contain a number of features that enable the detection of errors at both the radio frequency and the logical communication layers and in the identity discovery algorithm itself. These features enable robust discovery and communications in at least ideal communication situations and varying levels of protection as the level of communication interference increases. Ghost reads occur due to a reader's incorrect interpretation of the signals received from the tags. High reader density environments exacerbate these problems, particularly for protocols that retrieve only a portion of the tag identifier at a time; therefore, the reader implementations, specifically the extent to which error detection features are utilized, is the primary determiner of the probability that Ghost reads will occur for a particular protocol.

The key protocol features that enable the detection of tag communication errors leading to Ghost reads are contained within the identity discovery algorithm, often referred to as an anti-collision algorithm, the framing, or packetizing, of the tag communications, the radio frequency signaling utilized, including the encoding and modulation schemes, and the error detection, e.g., checksum, afforded the data communicated by the tag. We will examine each of these key features in turn, beginning with the identity discovery algorithm.

3. Identity Discover Algorithm

At its logical core, the protocol specifies a series of functions implemented on the tag. An identity discovery algorithm uses those functions to retrieve the stored on-tag identifier. The primary challenges with the design of the discovery algorithm, and hence the on-tag functionality, are the graceful recovery from communication errors, the proper management of multiple communicating tags, and the ability to successfully communicate in high reader density environments. The inability of tags to hear or communicate with other tags and the mix of relatively weak and relatively strong tag communication signals also must be accounted for in the design of the discovery algorithm. Each protocol addresses these challenges in its own fashion.

The EPCglobal Class 1 Generation 2 UHF (C1G2) protocol standard [6] utilizes multiple key algorithmic features to meet the primary design challenges of the discovery algorithm. We will examine each of these key features in turn. At its core, the C1G2 discovery algorithm uses a variant of the slotted Aloha anti-collision algorithm referred to here as the Q-Algorithm. The Q-Algorithm allows a reader to control probabilistically the number of tags that will communicate simultaneously in the first phase of the discovery algorithm. In this first phase, the reader communicates a response probability to tags in its communication zone. Given the response probability, a subset of the tags will respond to the reader, each with a probabilistically unique 16-bit random number (RN16) where each is referred to as the *handle* of the responding tag.

In the second phase of the algorithm, the reader responds to the tags with a handle that it received and was able to decode. This *preacknowledge* communication selects, with high probability, from those tags that responded in phase one, a single tag. The selected tags from phase two respond with their complete EPC identifier memory contents (which includes 16 Protocol Control (PC) bits, the object identifier, and a checksum) to complete phase three of the algorithm.

There is a 1 in 4,294,967,296 chance that two tags will respond with the same RN16; therefore, the preacknowledge feature effectively singulates a tag for communication, efficiently manages a large tag population, and gracefully recovers from tag-to-reader communication errors. Tags that do not receive a positive acknowledgment return to a safe, unidentified, state and do not interfere in the remaining phases of the identification process.

The preacknowledge feature addresses the potential disparities in tag communication strengths and minimizes the noise caused by tag-on-tag interference. By decreasing the noise in the communication channel, especially the coherent noise caused by multiple tags responding simultaneously, it is easier for the reader to decode correctly the communicated messages.

If the reader has difficulty decoding a tag's identifier communication in phase three, the C1G2 protocol enables a reader to request that the tag repeat its communication any number of

times. The confidence in a number can be increased to arbitrarily high values by repeatedly requesting the tag to respond with its identifier at the cost of decreasing its tag identification rate. This feature also enables the reader to determine a tag's identifier without ever receiving an error free communication from the tag. This can be achieved by comparing multiple responses from the tag and selecting the most common values that can be decoded. While technically possible, this approach does have the possibility of generating Ghost reads if a sufficiently high confidence level is not achieved.

By communicating the entire identifier within a single message, the protocol operates efficiently in high reader density environments. A single message containing the entire identifier eliminates the possibility that identifier portions from more than one tag are erroneously combined, eliminating one possible source of Ghost reads.

Furthermore, the communicated PC bits contain a length value that indicates the total number of bits communicated by the tag. This length value can be utilized by the reader to verify that the correct number of bits are decoded from the communication signal.

In contrast to the C1G2 protocol, the EPCglobal Class 1 Generation 1 UHF (C1G1) protocol standard [5] utilizes two principal features to meet the primary design challenges of the discovery algorithm, no on-tag state and tree-based identity discovery. Each command within the C1G1 protocol contains all of the state information required for that command. Thus, the C1G1 protocol assumes that the tag has no memory of prior commands.

The C1G1 algorithm utilizes a tree-based identity discovery algorithm. An oct-tree search command (Ping) retrieves three bits of data from the on-tag identifier memory of all tags in its communication field. The successive retrieval of identity memory bits enables the reader to effectively singulate a tag; however, the successive retrieval of identity memory bits requires that the reader communicate more and more information with each *Ping* command if it is to ensure that responding tags have the expected memory contents. When the reader believes that it has retrieved a sufficient number of memory bits to uniquely communicate with a tag, it requests, through a *Scroll* command, that the tag communicates its entire identifier memory which includes an object identifier and a checksum, a 16-bit CRC, on the identifier.

A piecewise variant of the C1G1 protocol that allows for a potential increase in the tag identification rate allows the reader to request that a tag it believes to be singulated communicate only those bits that the reader believes it does not currently know. A tag responding to such a request, communicates the contiguous portion of its EPC plus CRC beginning at the location directed by the reader. This approach allows for Ghost reads since a reader may logically join the EPC plus CRC segments from different tags, and the only protection against Ghost reads at that point is the CRC. We discuss the limitations of CRCs in protecting against errors in a later section.

In contrast to both the C1G2 and the C1G1 protocols, the EPCglobal Class 0 Generation 1 UHF (C0G1) protocol standard [4] utilizes a binary tree identification discovery algorithm that requires each tag to maintain its current state. The binary tree discovery algorithm retrieves one bit of tag memory at a time from tags within the reader's communication zone. Tags communicate using a frequency shift keying, or tonal communication, on a bit by bit basis. By closely tying the identity discovery algorithm and the tag communication signaling, the C0G1 protocol requires the reader to quickly and correctly confirm the communication from the tags. A decision is made by the reader on which direction to search down the binary search tree based upon the responses of the tags. In a

noise free environment, the identity discovery algorithm is able to efficiently identify tags. Communication noise including that generated by other readers can significantly degrade this performance by causing on-tag state to become out of sync with the reader's expectation of what the on-tag state is.

Protocols that retrieve an identifier plus checksum in separate and distinct communications have the potential for reader confusion in high reader density environments. It is possible for readers to hear tag communications from tags communicating with a different reader and for the reader to be communicating with different tags during the retrieval of each section of data. It is possible for a reader to believe it is communicating with a single tag when it is actually communicating with multiple tags or with a tag that is actually communicating with a different reader. Ghost reads are possible in many of these scenarios as communications from multiple tags are interpreted by the reader as communications from a single tag.

4. Communication Frames

The logical communication from the reader to the tag and from the tag to the reader is protected from errors by use of specified communication framing. The framing defines a sequence of required symbols interleaved with variable value symbols. The absence of required symbols or the presence of additional symbols within a frame indicate errors in the transmission. Variable length and variable value data, e.g., an EPC, should be protected individually.

Communication frames from the tag to the reader typically have a simple packet format that includes a Preamble, the data being communicated, and an End-of-Frame value. The packetization of the communication enables a reader to detect both the beginning and the end of the tag's communication, thereby providing error checks on the communication. As we discuss in the following section, the tag's communication signal is distorted by noise and other communications as it travels from the tag to the reader. If a tag's communication does not contain a Preamble and an End-of-Frame delimiter, the reader will need to sample noise to determine if communication has occurred. There is always at least a small probability that noise will look like valid communication, especially when decoding signals in a noisy communication environment.

In addition to the Preamble and End-of-Frame delimiters, a communication packet may contain additional delimiters to indicate the separation of one packet component from another. These Separation Symbols provide additional checkpoints that the reader may use to verify the correctness of the communications. The entire communication may also be protected by a CRC.

In protocols where the identifier is retrieved a portion at a time, the communication of each portion should be protected by a frame to enable at least the detection of errors. The CoG1 protocol retrieves the EPC and the CRC one bit at a time. The error detection for this one bit communicated from the tag is contained only in the signal itself. The tag signals for a binary one and for a binary zero create unique frequency spectrum signatures that contain sidebands symmetric about the reader's communication frequency. Robust communication occurs when both sidebands are present with exactly the same magnitude and the sequence of retrieved bits have roughly the same magnitudes. However, high tag and reader densities can have a significant impact on the sequence of magnitudes; therefore, the phases of the signals must also be examined. The total communication of the EPC is protected by use of a 16-bit CRC which is itself retrieved one bit at a time.

The C1G1 protocol allows for the EPC and the CRC to be retrieved either three bits at a time or as a continuous transmission from the tag. The three bit retrievals are not protected. The communication of the EPC using the *Scroll* command is protected by the use of a Preamble, a 16-bit CRC calculated over the EPC, and an End-of-Frame delimiter. When the EPC is communicated as a single message, the CRC is appended to the communication by the tag. Due to performance considerations, it is unlikely that a reader will utilize the three bit piecewise retrieval for the entire identifier. Rather, the three bit piecewise retrieval is used to attempt to singulate a tag before requesting that that tag communicate its entire EPC plus CRC in a single communication.

While the punctuation that a packet uses enables error detection checks on the decoded symbols, the detection of these symbols within prescribed time windows enables the reader to minimize the probability of sampling, and decoding, noise, or the signaling from a tag not communicating with that reader.

We evaluate the error detection capabilities of signals in the following section.

5. Signal Decoding

The reader perceives the communication from a tag as containing one of several possible waveforms. In the communication medium, the tag's communicated waveform may undergo some distortion. In most cases the distortion is caused by random processes, interactions with the environment, and possibly other tags. Because of their complexity and randomness, these sources of distortion may not be known precisely by the reader. The consequence is that a reader is no longer certain which of the possible waveforms was received. In all cases a signal is distorted as it travels through the medium and combines at the reader antenna so that the observable signal is itself a random process. Because of this, statistical methods are often employed to guide the analysis of the received signals. When analyzing the received signals, a reader must make decisions as to the presence or absence of a communicated waveform, the time of arrival of the waveform, its amplitude, phase, and other signal parameters.

The signal modulation, encoding, symbol timing parameters, communication frames, and the frame symbols provide multiple layers of decisions that the reader must make, and which the received signal must pass, to successfully decode a communication. The multiple layers of the signaling provide multiple checks on the decoding of the signal; thereby enabling a reader to selectively determine the probability of generating a Ghost read.

The signaling, or encoding, of symbols within the RF communication is robust against errors due to both the shape and the timings of the symbols. The symbol shapes are well defined within the protocol requiring specific ranges of timings for each feature of the symbol. Decoded symbol shapes that do not fit within the defined symbols denote communication errors. While there may be a broad range of possible timings for each symbol, the stream of contiguous symbols communicated by either a tag or a reader have only minor variations in both their timings and their amplitudes. Large variations in either timings or amplitudes, even if they are within the proscribed tolerances, can indicate the potential for a Ghost read.

The C1G2 protocol provides an array of signaling schemes that enable the reader to make educated decisions while decoding a received signal. C1G2 uses Amplitude Shift Keying (ASK) and/or Phase Shift Keying (PSK) modulation (as determined by the tag vendor) of the communicated symbols. Tags encode their back-

scattered data as either FM0 baseband or Miller modulation of a subcarrier. FM0 (bi-phase space) data encoding has memory, that is, the waveform communicated will depend on the value of the data communicated. Similarly, the Miller modulated subcarrier data encoding has memory. Memory in the encoding provides another mechanism that the reader can utilize to validate the correctness of the received message.

All communication from the tag to the reader utilizes a defined preamble to begin the communication frame. The FM0 encoding includes a specific data sequence and a single violation of the FM0 encoding. The Miller subcarrier preamble consists of a predefined sequence of data bits. The Miller modulated subcarrier encoding utilizes an End-of-Signaling value that consists of a single "dummy" data 1 bit. The FM0 encoding does not have an explicit End-of-Signaling value.

The signaling has specific timing parameters for when they must begin and their maximum duration.

6. Protecting the Data

Deployed RFID systems will communicate using a noisy radio frequency communication channel. All communications utilizing a noisy channel are subject to corruption, i.e., errors in the received message. Therefore, some form of protection against corruption must be afforded the communication. Since Ghost reads are the receipt by the reader of apparently valid data, we begin by examining how well that communicated data is protected. Protection in this context means enabling the reader to detect errors in the received data.

Error detection techniques are designed to enable the receiver of a message transmitted through the noisy channel to detect if the message has been corrupted during communication. Perhaps the simplest method for detecting errors is by calculating a value, referred to here as a checksum, that is a function of the message and appending it to the message prior to transmission. The receiver can utilize the same function to calculate the checksum and compare its calculated value with the received checksum value. Discrepancies in the calculated and received checksum values indicate errors in the received data.

The simplest checksum is a parity bit. A parity bit adds one additional bit to the transmission. The value of that bit is calculated such that the total number of binary one (1) values sent in the message is either even (even parity) or odd (odd parity). A parity bit enables the detection of odd numbers of bit errors only. Therefore, it is best used on very short messages, e.g., a single ASCII character, or in virtually noise free communication channels.

A checksum in the traditional sense is a summation of the data to be communicated. A traditional checksum is calculated by treating the data as a sequence of binary integers of length n and summing these integers, plus the carry, to yield an n-bit binary integer that is the checksum. A common choice for n is 16. Thus, the data is broken into 16-bit integers that are summed. Traditional checksums are easily calculated; unfortunately, they do not reliably detect all common errors. Multiple errors that "subtract" a value from one n-bit integer in the message and "add" that value to another n-bit integer elsewhere in the message have zero impact on the calculation of the traditional checksum. Such canceling errors are common in practice; therefore, traditional checksums do not provide adequate protection for most communications, including tag identification in RFID systems.

Fortunately, it is possible to improve the error detection capabilities of a checksum without increasing its length or significantly increasing its complexity. Instead of using addition to calculate the checksum, division may be used. A cyclic redundancy code (CRC) is a robust error detection technique that uses division to calculate a checksum that is appended to the end of the transmitted data. CRC algorithms treat the message data as a single number. This large value is divided by a fixed number called the CRC polynomial or generator polynomial. The remainder of this division is the checksum. The sending node calculates a CRC over the message to be transmitted and appends the resulting CRC to the message transmission. The receiving node calculates the CRC over the received message and compares its calculation with the received CRC. Alternatively, the receiving node divides the received message plus CRC by the generator polynomial. A remainder of zero indicates the absence of detected errors.

The CRC algorithm achieves robustness by using polynomial, as opposed to binary or decimal, arithmetic modulo 2. The use of polynomial arithmetic in calculating the CRC enables a 16-bit CRC to detect all single bit errors, all two bit errors, all odd bit errors, and all contiguous burst errors less than 17 bits in length. Many standard CRC generator polynomials have been defined. The 16-bit CRC-CCITT ensures detection of 99.998% of all possible errors [3]. Consequently, a 16-bit CRC is commonly used in data communications for data lengths up to 4 kilobytes [3].

The EPCglobal UHF Generation 1 Class 0 and Class 1 and UHF Generation 2 Class 1 protocols all utilize a 16-bit CRC to protect the communication of the object identifier stored on the tag [4][5][6]. Thus, the data, i.e., the identifiers, communicated by the tags is well protected *assuming* that the reader utilizes sound wireless communication techniques, also referred to as good radio design, in decoding the tags' communication signals.

A poor radio design will cause the reader, on occasion, to decode noise into a random message. Recall that a message is composed of data plus the CRC. A random 16-bit number has a 1 in 65,536 (1 in 216) chance of being a valid CRC for the data. Therefore, a good radio design must be used to virtually eliminate the probability of Ghost reads actually occurring.

7. Tag Memory Errors

One of the basic assumptions of a protocol is with regards to the tag memory capabilities. Tag memory, particularly low-cost passive tag memory, is very limited in size and capabilities. The protocols explicitly consider this limitation in their definitions of tag memory; thereby, allowing for the simplest and lowest cost tag memory implementations. Since the protocol defines only the logical interface to the memory, more complex memory implementations with error detection and correction capabilities are possible. However, such memory systems are expensive to implement and would typically be used only in ultra-high reliability applications (such as within spacecraft) that can afford the extra cost.

Since the protocols are typically designed to allow for the simplest and lowest cost memory implementations, there is often no explicit error detection or correction mechanism defined for the memory. However, some protocols, such as the EPC Generation 1 Class 0 (C0G1) and Class 1 (C1G1) protocols [4] [5], require the precalculation and storing in memory of the error detection mechanism, i.e., a CRC, used to protect the on-tag data. The CRC is calculated over the object identifier stored on the tag and explicitly stored in nonvolatile memory with the object identifier. These precalculated CRCs enable the detection of memory errors that

may occur after the data is correctly written to the tag memory and the detection of errors in the data received by the reader. Thus, by storing the CRC within memory, both memory errors and data communication errors may be detected.

The EPC Generation 2 Class 1 (C1G2) protocol [6], in contrast, does not explicitly store the CRC or utilize another error detection method for memory. Instead it relies upon the stability of the memory with its very low probability of errors to maintain the written object identifier. A CRC is calculated dynamically and used to protect the tag's data communication with the reader. Given the very small probability of memory errors it is sufficient to protect the communication but not the memory.

While many shortcomings of the forward and reverse communication channels can cause ghost reads, memory errors too can yield "apparent" Ghost reads.An apparent Ghost read yields the actual contents of memory; however, the memory contents have been unintentionally changed due to some natural phenomena. Since an apparent Ghost read yields the actual contents of memory, it is repeatable. Consequently, an apparent Ghost read has a potentially larger impact on the information system than does an actual Ghost read. The question that needs to be resolved is, therefore, "What is the probability of an apparent Ghost read?" or more accurately, "What is the probability of an unintentional change in the memory contents of an RFID tag?"

Passive RFID tags store their identifier(s) in non-volatile memories, either read-only memory, such as mask programmed memory, or field programmable memory, such as EEPROM. Once written, either in the manufacturing process or in the field, memory errors have been found to be rare in practice for non-volatile memories. The primary causes of memory errors are physical degradation or destruction of the memory or cells within the memory and radiation incident upon the memory.

True read-only memory, such as mask programmed memory or laser programmed memory, has an unlimited number of read cycles and is robust against physical errors. The primary causes of physical failure of either the memory or a cell within the memory are metal migration, an effect that causes open circuits in metal lines due to excessive currents in those lines, and physical impact and destruction. The silicon implementation can be designed to be robust against metal migration, thereby, virtually eliminating this type of failure. Furthermore, these memories are robust against radiation. Therefore, only the physical destruction of the chip must be protected against to protect the memory contents of read-only memories.

The dominant field programmable non-volatile memories today are EEPROM and Flash memory. Both EEPROM and Flash memory are electrically erasable and writable, and they are both traditionally based upon a device known as floatinggate MOS transistor (FGMOS). In current technologies, charge trapping is extremely reliable with retention times in excess of ten years. These memories have an unlimited number of read cycles, but a limited number of erase and rewrite cycles (typically on the order of 10^6 rewrites are possible).

Flash memories utilize a high voltage that is impressed on thin, fragile gate oxides to erase the memory. Today's memory devices contain internal charge pump circuitry that increases the applied voltage, often 3.3 V or 5 V, to as high as 28 volts for the purposes of erasure and 12 volts for the purposes of writing to a cell. Given the minimal thickness of the gate oxide (often between 30 and 100 Angstroms thick), the applied voltage is very stressful to the memory cell, and it will eventually lead to the breakdown and ruination of the cell if not the device. Current FGMOS-based memories can withstand more than 10^6 rewrites and feature retention times exceeding ten years [7].

Undue electrical abuse of the memory can decrease both the retention time and the life of the memory. Stress induced leakage current (SILC) is responsible for the floating gate charge loss phenomena. SILC is strongly voltage dependent. Therefore, induced voltages and currents on the tag, while not sufficient to destroy the tag, may be sufficient over time to induce a change in a stored value of a memory cell or destroy that cell altogether.

Non-volatile memory devices such as EEPROM and Flash memory are radiation resistant, but they can still suffer from soft errors caused by radiation. These memories can typically withstand up to 30 kRads of radiation without changing the value of a cell. Consequently, only a direct hit from a radiation particle can cause a memory value to change. The probability of a naturally occurring radiation particle hitting an EEPROM or Flash memory cell on a 128-bit RFID tag and changing the value of that cell is less than one in one billion [8]. Recent research has shown that the primary cause of failure due to radiation is the failure of the charge pump [9]. Thus, the memory will typically become inaccessible before its stored values are changed due to radiation.

Given the long retention times and low probability of radiation induced soft memory errors, apparent Ghost reads will occur with a very low frequency in a ubiquitous RFID system. Therefore, we need to be concerned with minimizing the actual Ghost reads that occur in an RFID system.

8. Reader implementations

All good protocols are designed to provide communication robustness in a noisy environment. It is incumbent upon the reader implementation to utilize these features to ensure that erroneous data, e.g., Ghost reads, are not reported to the information system. Some reader implementations will take advantage of statistically probable outcomes to improve their performance, i.e., identification rate, typically at the expense of increasing the probability of Ghost reads. In noisy environments, such statistical gambles are required to achieve the highest performance. However, such gambles also open the door for possible Ghost reads. We examine some of the gambles that may be taken by a reader implementation.

Reader implementations will attempt to improve performance in non-ideal environments by minimizing or ignoring the error detection capabilities inherent in the protocols, particularly the signaling. For example, while communicating with CoG1 compliant tags, a reader may aggressively identify a bit communicated from a tag even if only one sideband is decodable. In noisy environments it will often be the case that one sideband is obscured by noise while the other is decodable. Therefore, aggressively decoding only a single sideband will improve performance. However, without the error check inherent in receiving two sidebands of equal magnitude, there is no certainty that the decoded sideband is not from a tag controlled by another reader or simply noise in the environment. This technique places great emphasis on the CRC to catch errors, which, as we have seen has a high probability of allowing Ghost reads to be reported as valid reads. Unfortunately, it is possible for a reader that is decoding one sideband only to “receive” an EPC that passes its CRC check but is in fact a Ghost read.

Some reader implementations may attempt to decode signals in extremely noisy environments and very weak signals that are at or near the noise floor. In such cases, the reader may in fact be pulling random numbers from the noise. Symbol timings and structure offer little error checking capabilities in such environments, as the symbols are very distorted by the noise. The required symbols in the communication frame offer some protection; however, reader implementations may ignore these required symbols that, in a tag response containing its EPC and CRC, amount to a simple preamble only. As indicated earlier, the retrieval of random numbers has a 1 in 65,536 (1 in 2^{16}) chance of passing the CRC. This leads to one Ghost read every three minutes for a reader "reading" noise at a rate of 400 tags per second.

Recognizing that a completely random number is not a properly formed identifier, reader implementations may self direct along a predetermined identifier path. Thus, if there is any structure in the identifier, such as exists in the EPC, the reader may follow this structure in a predetermined fashion. For the bit-by-bit retrieval method of the C0G1 protocol, self direction within a multiple reader environment eliminates the error detection capabilities of the CRC. Thus, Ghost reads will be reported almost continuously as noise is decoded as valid tag communications.

For contiguous communications such as those required in the C1G1 and C1G2 protocols, the symbols may be decoded in the predetermined fashion for all portions of the EPC and CRC except the serial number. Fortunately, a simple consistency check on the symbols can quickly eliminate those numbers whose symbols were not similar in timings and shape over the course of the entire message. However, not all errors can be detected in high noise environments.

The communication from a single tag probably will have great consistency in its amplitude and its symbol timings over the course of a single communication. Minor variations may exist due to motion or noise in the environment. Some reader implementations will ignore all variations in the signal strengths or symbol timings and rely upon the CRC to detect any errors in the resulting identifier. When consistency checks are not used and the reader self-directs the decoding, Ghost reads will result.

For the C0G1 systems, a consistency check on the symbols is less reliable in weeding out potential Ghost reads. When combined with single sideband decoding and high reader density environments, a consistency check would aggressively weed out valid tag reads while still allowing for the potential of Ghost reads. Self directed decodings with the C0G1 protocol in high reader density environments will yield Ghost reads.

For the C1G2 systems, the use of all error detection capabilities within the protocol will enable high performance (high identification rates) while simultaneously virtually eliminating the appearance of Ghost reads.

9. Conclusions

Ghost reads are a potentially large impediment to the use of RFID systems for large-scale object identification. Ghost reads cause data errors to be automatically entered into the information systems; therefore, they must be eliminated to the greatest extent possible. Tag memory errors will occur on very rare occasions and will appear to the system as apparent Ghost reads. Since the contents of the tag's memory must have changed to create an apparent Ghost read, it is an easily repeatable event. As memory errors are rare and isolated events, the appearance of apparent Ghost reads can be quickly identified and resolved. Similarly, the

rare appearance of an actual Ghost read can be easily managed. Conversely, the widespread appearance of Ghost reads can paralyze an information system.

Ghost reads result from failures in the communication from tags to readers. These failures are due primarily to reader implementations that do not fully utilize the error detection capabilities of a protocol. High reader density environments will exacerbate the appearance of Ghost reads. Protocols that retrieve data in a piecewise unprotected fashion are particularly susceptible to Ghost reads in high reader density environments; particularly when reader implementations that do not fully utilize the error detection features of the protocol are used.

Fortunately, Ghost reads can be effectively eliminated by eliminating the reader implementation shortcuts that are used within the reader. The C1G2 protocol will also minimize the appearance of Ghost reads since EPCs are retrieved from tags only by a contiguous tag transmission that includes a 16-bit CRC. Consistency checks will eliminate nearly all Ghost reads and identify those communications that are potential Ghost reads.

AUTO-ID LABS

References

[1] D. W. Engels and S. E. Sarma, “The reader colision problem,” in Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, October 2002.

[2] J. Waldrop, D. W. Engels, and S. E. Sarma, “Colorwave: An anticollision algorithm for the reader collision problem,” in IEEE International Conference on Communications (ICC03), May 2003.

[3] A. S. Tanenbaum, Computer Networks. Prentice-Hall, 1981.

[4] EPCglobal, Inc., “860MHz960MHz Class 0 Radio Frequency Identification Tag Radio Frequency and Logical Communication Interface Specification, EPCglobal Standard,” 2003.

[5] ——, “860MHz960MHz Class I Generation 1 Radio Frequency Identification Tag Radio Frequency and Logical Communication Interface Specification, EPCglobal Standard,” 2003.

[6] ——, “EPC Radio Frequency identity Protocols Class 1 Generation 2 UHF RFID Protocol for Communications at 860MHz960MHz, EPCglobal Standard,” 2004.

[7] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, Flash Memories. Kluwer Academic Publishers, 1999.

[8] J. Ziegler, “Terrestrial cosmic rays,” IBM Journal of Research Developments, vol. 40, no. 1, January 1996.

[9] D. Nguyen, S. Guertin, G. Swift, J. Coss, and A. Johnston, “Radiation effects on advanced flash memories,” in Proceedings of the 1999 IEEE Nuclear and Space Radiation Effects Conference, July 1999.

Acknowledgement

The author would like to thank the Technical Steering Committee of EPCglobal, Inc., particularly Steve Rehling and Chris Diorio, and Professor Peter Cole, Mun Leng, and Kin Seong from the University of Adelaide who provided valuable input and comments that have contributed to improving the content and exposition of this paper. This work was supported by EPCglobal, Inc.

AUTO–ID LABS

Daniel W. Engels

Dr. Daniel W. Engels is the Director of Research of the Auto-ID Labs of the Massachusetts Institute of Technology, a groundbreaking research center that explores applications and develops technologies for ubiquitous intelligent object networks. He is one of the principal architects of the Networked Physical World EPC System, the foundation of the Internet of Things, developed under the Auto-ID Center and licensed to EPCglobal Inc. Dr. Engels is a member of the Technical Steering Committee under EPCglobal, Inc. and led the development of RFID protocols under the Auto-ID Center. Dr. Engels received his Bachelors of Science degree from the University at Buffalo, his Masters of Science degree from the University of California, Berkeley, and his Ph.D. from the Massachusetts Institute of Technology. Dr. Engels Masters thesis was in the area of real-time operating systems and computer-aided design, and his Ph.D. thesis was in the areas of scheduling complexity theory and embedded system design. His current research is in the areas of radio frequency identification (RFID) system design, RFID protocol design, applications of RFID, RFID antennae design, sensor networks, and intelligent objects. He has over 25 publications in RFID, RFID applications, security, embedded computing, and computer-aided design.