RFID Tag Counting Protocol

John Emmanuel Palagiri
Electrical and Computer Engineering
North Carolina State University
Raleigh, United States
jpalagi@ncsu.edu

Shanmukha Jaya Harsha Alluri
Computer Networking
North Carolina State University
Raleigh, United States
salluri@ncsu.edu

Pabba Akhil
Computer Networking
North Carolina State University
Raleigh, United States
apabba@ncsu.edu

***Abstract*—Counting the number of radio frequency identification (RFID) tags, namely RFID counting, is needed by a wide array of important wireless applications. Motivated by its paramount practical importance, researchers have developed an impressive arsenal of techniques to improve the performance of RFID counting (i.e., to reduce the time needed to do the counting). Existing tag counting protocols, such as Aloha variants use TDMA, in which the tags are asked to respond in a timeslot that is given to the tag in the initial phase of counting. By varying the number of slots per round, the Aloha protocols aim to get the accurate count of the tags present in the counting scenario. In this paper, we present a new way of counting tags, based on the bits in tag ID. This technique is based on CDMA, which inherently avoids collisions while sharing the bandwidth available with all the users at a single point of time. We believe that our proposed protocol outperforms the present protocols in terms of time efficiency.**

***Index Terms*—RFID Tag Counting Protocols, TDMA, CDMA, Aloha Variants.**

I. Introduction

RADIO frequency identification (RFID) technology uses RFID tags and RFID readers (or simply called tags and readers) to monitor objects in the physical world. A tag is a low-cost microchip that can be attached to an object. It can store some information (including a unique ID) and can communicate with a reader through a wireless channel. Over the past decade, RFID technology has enjoyed significant growth. With six billion tags sold in 2013, RFID technology has by now impacted applications ranging from inventory control, supply chain management, to people tracking [17]. A common basic functionality needed by many of these applications is RFID counting—to count the number of tags and thus the number of tagged objects in a certain physical area. For example:

- Wal-Mart puts tags on individual clothes. Here, RFID counting provides information about the sales trend and speeds up the restocking process.
- Purdue Pharma has tagged millions of its tablet bottles. Here, RFID counting ensures the right amount of its products are passing through its manufacturing, packaging, and shipping process.
- Many events distribute RFID wristbands to their visitors. Here, RFID counting helps reveal the number of people around.

Typically, an RFID system comprises:

- One or more tags. These include an IC-chip, an antenna and are attached to the objects to count or identify. Tags can be active (battery operated) or passive (no battery). Because passive tags are activated using coupled power originated from the reader, the latter has lower coverage.
- A reader/interrogator. This device is made up of an RF module, a control unit, and one or more antennas. It offers a bidirectional communication between the tags and the reader.
- A data processing subsystem. Connected to the reader, allows for the storage and further processing of the data information of identified tags into a database.

The coexistence of various tags sharing the communication channel leads to a unique problem known as the tag collision problem. The reader detects the collisions by monitoring the replies that are back-scattered by the tags. These replies usually contain the tag ID of the replying tag. A collision makes these messages unreadable by the reader, thus forcing the tags to re-transmit tag IDs, which results in a loss of bandwidth, an increase of power consumption, and a large delay in the identification process. To face this problem an anti-collision protocol is needed.

In literature, several proposed protocols have been reported, and they can be classified in aloha based, tree based and hybrid protocols. These protocols are divided into two categories, Probabilistic and Deterministic [1].

Probabilistic protocols aim to get an accurate count, while Deterministic protocols aim to get an exact count of the tags. The Aloha based protocols count the tags by giving the random numbers to the tags at the beginning of the counting. The tag will wait for its turn to reply based on the random number that is assigned to it. These protocols suffer from the tag starvation problem [19], in which a tag may not be read in a reading cycle.

Then comes the Tree based protocols, these Tree based protocols are deterministic and provide simple tag designs, one of such examples is the Query Tree (QT) protocol [18]. These protocols read all the tags in the interrogation zone on each cycle.

The Hybrid protocols, these protocols are specifically designed to avoid the problems of aloha and tree-based protocols at the expense of a complex reader and tag designs.

In this paper, we propose a fresh way of counting the tags by using the CDMA (Code Division Multiple Access) to know the tag ID. We say it a fresh way because from our literature readings we did not find any research in employing CDMA technique to RFID tag counting.

In the subsequent sections, we give an overview of the present protocols in Related Work (Section II) and in Section III we present our Idea and explain in detail about its functionality and in section IV we will present the plan of work and finally section V for references.

II. Related Work

In this section, an overview of the present anti-collision protocols is presented. In RFID systems, TDMA procedures are the most used techniques in RFID and they have the largest group of anti-collision methods. These can be categorized into Aloha-based protocols which are probabilistic, Tree-based protocols which are deterministic, and hybrid protocols which are a mixture of the previous ones.

With our literature readings on anti-collision protocols, we review them briefly in this section and present the problem in these protocols. Afterward, we present the idea of CDMA and how that can be employed in RFID tag counting. Here are some of the protocols that are existing in the present: Slotted-Aloha, Frame-Slotted-Aloha, first non-empty slot-based estimation [4], probabilistic estimating tree [5], average run-based estimation [6], Query Tree, Tree Splitting, Binary Search.

A. Aloha Based Protocols

Aloha is a protocol where a user can transmit data whenever they have data or when they are asked to transmit, this protocol here has a very high chance of collisions as there can always be multiple users transmitting on the same channel. As a result of that problem an improvised method of that protocol is introduced which is the Slotted-Aloha, which introduces the slot concept. A slot is a period during which the reader sends a command and the tags respond to the reader. So, slotted-Aloha divides time into slots and as a result, the users are allowed to transmit only in the allocated slots reducing the collision problem and as a result in improving its throughput. If collision occurs, tags re-transmit after some random delay. Since the collisions happen at boundaries of the slot only, there will be no partial collisions [17].

Later, Framed-slotted-Aloha (FSA) protocol is developed, in which all nodes must respond to choosing a slot into a fixed length frame (a group of slots). In pure Aloha and Slotted Aloha based systems, a tag that has high response rate will collide frequently with potentially valid responses from the other tags. As the throughput of the FSA decreases with the increase of the total amount of nodes, a Dynamic-framed-slotted-Aloha (DFSA) is developed [17]. These are Frame slotted aloha protocols with variable frame sizes. This protocol changes the length of the frame dynamically using an estimator to adjust the frame size. Some protocols like I-Code [13] change the frame size at the end of the last frame slot, and other algorithms, as the EPC C1G2 Slot Counter [11], adjust the frame size after a slot transmission. DFSA operates in multiple rounds and it can also include the early-end feature. In each read round, the reader uses a tag estimation function that is used to vary the frame size. This tag estimation function takes the feedback from the reader's frame and calculates the number of tags. One of the DFSA variants limitation is that the frame size is bounded to a maximum value of 256 or 512. And when the number of tags value goes beyond this value, persistent collisions becomes a main issue.

To end this, another version of DFSA was proposed, called enhanced-DFSA or EDFSA. Here the tags are divided into M groups and if the tag population is greater than the maximum frame size that is available. For different tag ranges, we have different values of M. The value of M is one when the number of tags is less than 354. However, when the number of tags increase, the modulo operation comes into effect, which divides the responding tags into M groups. Then the reader reads the tags on a group-by-group basis.

As cited earlier, the tag starvation problem affects probabilistic algorithms, that is, a tag that may not be correctly read during a reading cycle. Estimation involves some disadvantages: an increase in the computational cost of the reader and the tag; an error that degrades the efficiency; and lastly, an initial frame length cannot be set according to the estimated number of tags.

B. Tree Based Protocols

The Tree based protocols are deterministic in nature, as a result, they measure the exact count of the tags. These protocols usually have a simple design and work well with a uniform set of tags but are slower than Aloha-based protocols [17].

These Tree based protocols can be categorized into Tree Splitting (TS), Query Tree (QT), Collision Tree(CT) and Optimal Query Tree[9]. In Tree Splitting (TS) technique, the clock is synchronized with the subset of tags and tries to get the response of the tags in slots to estimate the number of tags. TS protocols operate by splitting the responding tags into various subsets using a random number generator.

In Query Tree (QT) based techniques [14], each tag has a prefix matching circuit. The reader transmits a query q and ifthe tag matches the bits with the input of the reader, then the tag replies else it will not so in this way the exact tag ID data and the count can be found easily. Collisions occur when more than one tag has same prefix. In this case, the reader forms a new query in which it appends q with a binary 0 or 1. This reading process is repeated by the reader using the augmented query.

In Optimal Query Tree, the tree divides all the tags in the interrogation area into small tags sets in order to reduce the number of collisions at the beginning of the Identification process. This protocol uses three main approaches: Bit estimation, an optimal partition, and a query tracking tree. Bit estimation - With this, the reader using bit-tracking technology can estimate the number of users in the interrogation area with a small deviation.

Optimal Partition - This divides all the tags in the interrogation area into multiple sets with initial queries. Query Tracking Tree - This splits the collided tags into two subsets using the first collided bit of the tag's response. This procedure is continued until there are no more queries left in the stack.

Lastly, the Collision Tree(CT)[9]. The collision tree (CT) protocol is an improvement of QT protocol which uses bit-tracking technology in order to find which bits have collided. Here in this protocol, with the bit tracking technology it can trace a collision to the individual bits and gets the correct bits successfully. The basic feature of this protocol is to decrease the collision slots and eliminate the Idle slots. The performance of CT is very dependent on the total number of tags in the interrogation area.

A collision is observed only if two tags respond with different bit values. Moreover, the reader must specify the bit position it wants to read. The reader uses a stack to store tags' position on the tree, while a tag has a counter to record the depth of the reader's stack. Based on this counter value, a tag determines whether it is in the transmit or wait state. In other words, a counter value of zero moves a tag into the transmit state.

Otherwise, the tag enters the wait state. Once a tag is identified, it enters the sleep state. These tree-based protocols can find which bits are colliding in a collision slot.

III. Model

A. Background

The above-discussed protocols are time division based and takes a lot of time in estimating the number of tags. To reduce the time taken we can use CDMA, in which instead of time splitting of the available bandwidth, codes are given to the users to encode and decode the data. Data from one user will be anonymous to the other. These features make CDMA perfect in the scenarios where we must provide access to many users at the same time without any collisions. As per our readings, there is no prior work in defining the CDMA protocols in RFID tag counting.

CDMA gives the codes to the users at the start of the communication. These codes are called as spread codes. Spread codes are used to spread the data from the user into a wider bandwidth. CDMA gives the spread codes to the users from the Walsh matrix, which generates the codes that are orthogonal to each other. These Walsh codes contain good auto-correlation and worse cross-correlation properties [15], which reduce the Inter Signal Interference (ISI). By reducing the ISI, CDMA can assign the available spectrum to the users that are in the communication link without any collisions.

B. Model Working

Walsh codes are generated from the Hadamard matrix. Hadamard matrix is a square array of plus and minus ones, +1, -1, whose rows and columns are mutually orthogonal. +1 is replaced by 0 and -1 is replaced by 1, for using these codes in encoding and decoding of the binary data. Fig.1 shows a 2x2 Hadamard matrix. Any higher order matrix can be constructed from the lower order by the formula shown in Fig.2. For example, Fig.3 shows a H4 higher order matrix. Each row in the Hadamard matrix corresponds to one Walsh code.Each row in the Hadamard matrix corresponds to one Walsh code. CDMA distributes these codes to the users. The users encode their data with the code assigned to them and transmit the data to the mobile base station. Let us work with an example to see how the CDMA works.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Fig. 1. 2*2 Hadamard Matrix

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & \overline{H_N} \end{bmatrix}$$

Fig. 2. Higher order matrix construction

Consider a scenario where we have 4 users and each user must transmit 2 bits of data. Each user will be assigned a Walsh code that is generated from the Hadamard matrix. ('+1' is replaced by '0' and '-1' is replaced by '1')

Data – Spread Code

User 1 00 – 0000
User 2 01 – 0101
User 3 10 – 0011
User 4 11 – 0110

After the user gets his code, the data that must be transmitted by the user should be encoded by the code assigned to it. This encoding scheme is an XOR function. The 2 bits in the data field are spread across the 4-bit code received by the user and XOR operation is performed. The first bit is repeated 4 times and XOR operation is performed. The same is operated on the second bit. The result of 8-bits will be transmitted to the base station.

After XOR operation -

00 – 0000 0000
01 -– 0101 1010
10 — 1100 0011
11 — 1001 1001

$$\mathbf{H_4} = \mathbf{H_2} \times \mathbf{H_2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Fig. 3. H4 matrix

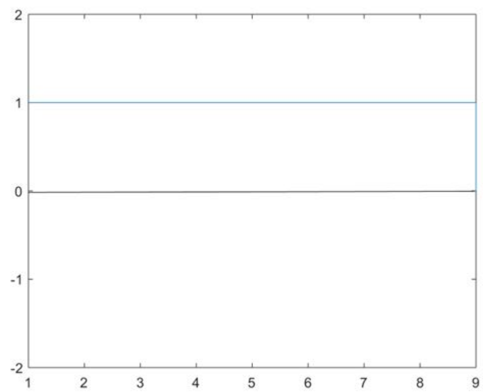

Fig. 4. Plot for ’00’ signal

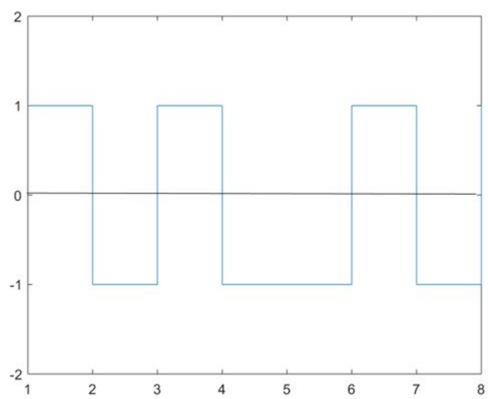

Fig. 5. Plot for ’01’ signal

When the base station receives the data from the different users at the same time, the result will be the mixed signal of all the users. To distinguish between the 0’s and 1’s, the reader relates them to +1V and -1V respectively.

Let’s plot them according to the signal levels and add them to get the multiplexed signal of the users at the base station.

The plots for ‘00’,’01’,’10’,’11’ signals are shown in Fig.4, Fig.5, Fig.6, Fig.7 respectively.

And the multiplexed signal result will be the signal shown in Fig.8.

In sequence: 0,0,4,0,0,4,0,0.

The multiplexed signal will be decoded using the codes that are distributed to the users. Fig.9 displays the decoding process in detail.

Since 1 data bit is spread across 4 bits, we add the 4 bits and divide them by 4 to get the users data back.

CDMA works on this technique to decode the data of different users at the same time. This is an efficient way to divide the available spectrum among the users simultaneously. CDMA uses 64-bit spread codes to encode and decode the data. However, we cannot use the 64-bits to get the tag ID, as the computational power of the tag is defined by the signal strength it receives from the reader [8].

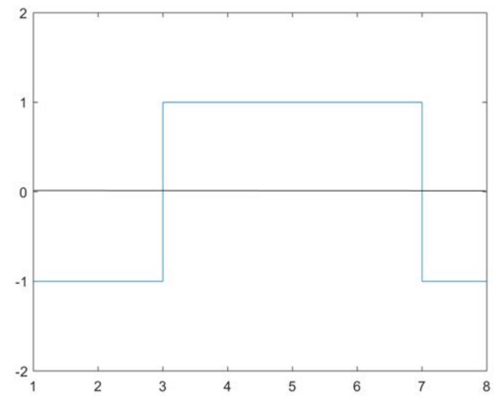

Fig. 6. Plot for ’10’ signal

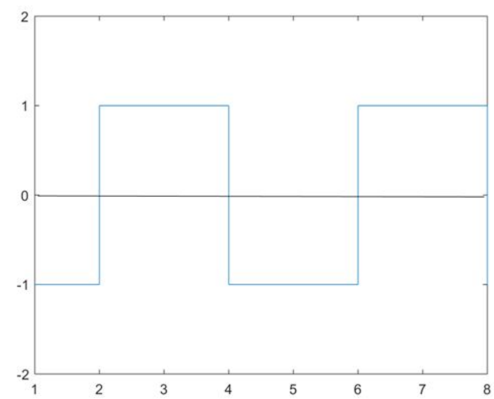

Fig. 7. Plot for ’11’ signal

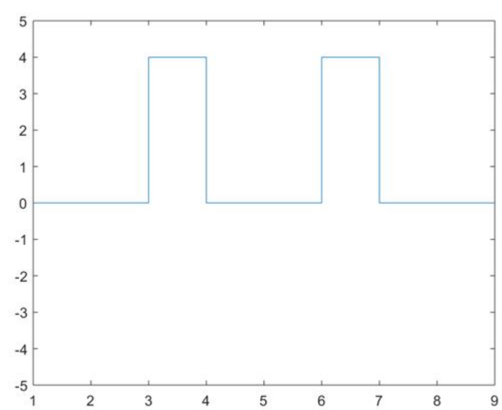

Fig. 8. Resulting multiplexed signal

0 0 4 0	0 4 0 0	(Result)
0 1 1 0	0 1 1 0	(Code for User 4)
0 0 -4 0	0 -4 0 0	

Fig. 9. decoding the multiplexed signal

The above technique can be applied to the RFID tag counting problem, but with variations because of the energy constraints that are posed by the passive tags. The limitations would be in assigning such a big 64-bit spread code to the tags. Optimizations include using the same code between tags that contain the same data bits in the specified range by the reader. But, using the same code with different tags poses a problem of collision. Surprisingly, the collisions can be detected, because the same data that is transmitted by the tags will be added to the multiplexed signal. So, the multiplexed signal will contain same data amplified. Measuring this amplification factor will give us the number of tags that are transmitting the same data.

Let's repeat the above problem with 2 users that want to transmit the same data. Since, they have the same data, the codes given to these users must be the same.

Data – Spread Code

User 1 00 – 0000
User 2 01 — 0101
User 3 10 — 0011
User 4 11 -– 0110
User 5 11 – 0110

After XOR operation -

00 -– 0000 0000
01 — 0101 1010
10 — 1100 0011
11 — 1001 1001
11 — 1001 1001

The result at the reader will be -1,1,5,-1,-1,5,1,-1

-1 1 5 -1 -1 5 1 -1 (Result at the reader)
0 +1 +1 0 0 1 1 0 (Code for User 4,5)

-1 -1 -5 -1 -1 -5 -1 -1

-1-1-5-1 = -8/4 = -2 $\implies 2 \times (-1)$

$$-1-1-5-1 = -8/4 = -2 \implies 2 \times (-1)$$

The result will be 2 times the data bits, as we have received two responses from the tags with same data. Thus, we will be able to detect the responses from different tags containing the same range of bits that are encoded

IV. PROPOSED SCHEME

Consider there are n number of tags in the environment to read. Also consider that the tag ID is 8-bit long. The reader sends a clock synchronize signal to all the tags. Tags receive the signal and synchronize their clock with the readers clock. The reader then sends a spread code function to all the tags. This function generates the spread code based on the tag ID bits. The function checks for the range of bits in every tag and assigns the spread code based on the tag's bits. This means that two tags who has the same range of bits will be assigned with same spread code.

Now consider that the spread function checks for first 2 bits in the tag ID. Since 2 bits can form 4 different combinations the spread code will be 4 bits long. After knowing its spread code, the tags will perform the XOR operation with the first 2 bits in the tag. The result of 8 bits will be transmitted back to the reader by all the tags at the same time (since the clock signal is synchronized with the reader). The reader receives all the replies from the readers as a multiplexed signal. This signal will be decoded using the spread codes that are distributed to the tags. After this process the reader will come to know which range of bits are present in the counting scenario and how many tags are present for a specific combination. **Algorithm 1** provides the steps for this counting method.

Algorithm 1: Tag counting

- Send the clock synchronize signal to the tags.
- Generate 4 spread codes and assign them to four 2-bit combinations.
- (First 2 bits of tag ID == ith Combination) Result = (First 2 bits) XOR (ith spread code)
- Transmit the result to the reader.
- Calculate the number of tags replied for ith combination.
- Number of tags = Total replies in all 4 combinations.

To know the complete tag ID of all the tags present in the system, the above process must be repeated with the next range of bits. Since the reader knows which first 2 bits combination is present in the system, it uses this information in the spread function. Let's assume that 4 tags have replied with the combination 00 in their first 2 bits and 3 tags have replied with 10 combination. Now the reader must divide the time into slots based on the combination. Since, the reader knows that 4 tags have replied with 00 combination, the spread function transmitted to the tag's checks for the first two bits. In case of match, the spread code will be assigned for the next 2 bits and the XOR operation will be performed which is followed by the transmission of the result to the reader. If the first 2 bits don't match the tag will keep silent in this process and waits for its turn. The reader repeats this process recursively, until all the tags and their tag ID's are read.

V. COMPARING CDMA TO OTHER PROTOCOLS

In this paper, we would also like to compare certain factors like RTF/TTF, system cost, complexity, efficiency and the disadvantages of all the above-mentioned protocols[18], with the protocol that we are presenting. Firstly, comparing the Aloha-based protocols:

A. Slotted Aloha

- The slotted Aloha is a RTF following protocol, which means that the reader is the one that talks first and not the tags. So being a slot-based protocol, the tag IDs will be transmitted in synchronous time slots trying to avoid collisions but it has its own disadvantages.
- Disadvantages: If the number of tags is too dense then the number of collisions will be increasing significantly and the reader requires a proper synchronization with the tags.
- The efficiency of the slotted Aloha protocols is only **36.8%** and the complexity of the system is also low, and as a result, the equipment needed for establishing this system is low [18].

B. Frame Slotted Aloha

- The FSA algorithm is an RTF following protocol as well and here the time is divided into frames and the tags will be transmitting only once per frame.
- Disadvantage: One of the major disadvantages is that it uses fixed frame size and does not change the size during the tag counting process resulting in a delay of the identification process.
- Another inconvenience with the FSA is that when the number of tags is low, and the frame size is big, the slots are wasted.
- Though the efficiency of FSA is only **36.8%**, as the complexity of the system is medium as a result the cost of the system is High compared to Slotted Aloha systems [18].

C. Dynamic Frame Slotted Aloha

- The DFSA is again an RTF following a protocol similar to both SA and FSA. The major advantage of it over the FSA is its dynamic nature to change the Frame size before every Frame.
- Disadvantage: A big disadvantage of this model is that it cannot move into the next frame at any time based on the situation of the collision without finishing the current frame.
- The efficiency of the DFSA algorithm is **42.6%** which is more than both SA and FSA protocols due to its dynamic nature. The complexity is also High as well resulting in the high cost of the system [18].

D. CDMA Technique

- Our proposed Idea uses CDMA and it is also an RTF following protocol, and here the system listens on multiple channels for the simultaneous responses of the RFID tags.
- Disadvantage: The disadvantage of the system is that, in a case where it has to obtain the Tag ID, the tags have to constantly perform the XOR operations for every iteration, as a result, this will overload tag with energy consumption.
- The efficiency cannot be given out as a number but theoretically, it will be the most effective system for the RFID counting as we were able to identify the count in a single go.
- The complexity of the system will be at the reader as it has to read data from multiple channels and keep a count. As a result, the cost of the system would be high due to the requirement of a complex reader.

Overall, looking at all the techniques for calculating the RFID tags, we can infer that all of the protocols mentioned here will only estimate the count but with the CDMA technique, we can get an exact count. Though the complexity and the cost of the system are high in CDMA technique, it gives out the exact count in a single round.

Next, comparing the Tree based techniques:

E. Query Tree

- In QT, the tags start transmitting only if the bits in ID matches the query sent out by the receiver.
- Disadvantage: Major disadvantage is that it produces a huge number of collisions in the beginning steps of the process.
- Due to the huge number of collisions in the beginning phase, the efficiency is only **34.6%**, the complexity is low, and the cost is very low [18].

F. Binary Search Tree

- In BS the reader transmits a serial number, and the tags having an equal or lower ID than the received serial number will respond to the request.
- Disadvantage: The disadvantage in this protocol is that the reader restarts the reading the process after a tag is identified.
- The efficiency of the system is only **30%** and also the cost of the system and the complexity of this model is also low [18].

G. Optimal Query Tree

- In this QT protocol, the reader performs an initial estimation and divides the tags into groups using Collision Tree protocol.
- Disadvantage: Very complex in nature, hard to be physically implemented.
- The efficiency is **61.4%**, which is the best among the tree-based protocols and as a result, the cost of the model is very high and so is the complexity [18].

H. Collision Tree

- The CT protocol is an improvement of QT protocol, which uses bit tracking to find which bits collided and where are they.
- Disadvantage: The problem here is, it wastes too many numbers of tag bits on every collision resulting in the increase of energy consumption by the reader during the process.

- As it is only an improvement from QT, the complexity of the system is only medium and so as the cost. But it gives out the efficiency of **50%** when compared to **34.6%** of the QT [18].

The major difference between our CDMA technique and the above-mentioned protocols is that unlike these Tree protocols, CDMA technique does not need the tag ID's knowledge completely to get the exact count of the system. The advantage of our idea is that we are able to get the tag count in a single round but for knowing the complete tag ID of the tags we can extend our algorithm by implementing the second phase similar to the query tree. In QT, the tag count is calculated after all the tags have been queried for their complete tag ID but having the knowledge about the number of tags before the second phase is clearly an advantage to us. Since the reader can listen on multiple channels, we will be querying the tags in time slots and can cross check the tags replies based on the count that we already have at the end of the first phase.

VI. FUTURE WORK

Simulating and comparing the presented technique with Aloha and Tree-based algorithms is our future work. We ensure that with time and resources, our idea can be simulated and compared with the present algorithms. We found that OMNET++ and NS(Network Simulator) are two software distributions that are available for conducting the simulations related to RFID tag counting problem.

VII. CONCLUSION

We have presented a novel idea of counting the tags in the ecosystem using CDMA techniques and comprehensively understood and compared with present algorithms in RFID tag counting problem. In general, two methods are used for identifying tags: Aloha and Tree. The key advantage of aloha protocols is dynamic adaptability to varying loads and low reader to tag commands. On the other hand, tree protocols promise deterministic identifications but require a high number of readers to tag commands. When it comes to tag counting, using the CDMA technique theoretically outperforms the present algorithms in the RFID industry by removing the collisions in the problem. When compared to tree-based protocols, the presented algorithm efficiently uses the replies from the tags and monitors the bits for knowing the exact tag ID of the tags.

However, a key limitation of our idea is the high cost and complexity at the reader. Looking at the trends of the industry, in the coming future, the RFIDs would be used in every major sector. Addressing the tag counting problem with better efficiency was always our motive and we have presented an algorithm which addresses this problem.

VIII. ACKNOWLEDGEMENT

We acknowledge and thank the support of our professor Dr. Khaled Harfoush, Associate Professor of Computer Science department at North Carolina State University.

REFERENCES

[1] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in Proc. ACM MobiCom, 2006, pp. 322–333.

[2] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using RFID tags," in Proc. IEEE INFOCOM, 2007, pp. 1217–1225.

[3] C. Qian, H. Ngan, Y. Liu, and L. Ni, "Cardinality estimation for large-scale RFID systems," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 9, pp. 1441–1454, Sep. 2011.

[4] H. Han et al., "Counting RFID tags efficiently and anonymously," in Proc. IEEE INFOCOM, 2010, pp. 1–9.

[5] Y. Zheng and M. Li, "PET: Probabilistic estimating tree for large-scale RFID estimation," IEEE Trans. Mobile Comput., vol. 11, no. 11, pp. 1763–1774, Nov. 2012.

[6] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in Proc. ACM MobiCom, 2012, pp. 365–376.

[7] Y. Zheng and M. Li, "ZOE: Fast cardinality estimation for large-scale RFID systems," in Proc. IEEE INFOCOM, 2013, pp. 908–916.

[8] Ziling Zhou, Binbin Chen and Haifeng Yu, "Understanding RFID Counting Protocols", IEEE/ACM Trans. Networking, vol. 24, no. 1, pp. 312–327.

[9] Hugo Landaluce, Asier Perallos, and Ignacio Angulo, "Managing the Number of Tag Bits Transmitted in a Bit-Tracking RFID Collision Resolution Protocol" Multidisciplinary Digital Publishing Institute (MDPI).

[10] Frits C. Schoute. "Dynamic Frame Length ALOHA". IEEE Transactions on Communications, COM-31(4):565-568, April 1983.

[11] "EPC radio-frequency identification protocols class-1 generation-2 RFID protocol for communications at 860 MHz-960 MHz Version 1.0.9." EPCglobal, Jan. 2005.

[12] Su Ryun Lee, Sung Don Joo and Chae Woo Lee. "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. pp 166-172, July 2005.

[13] Vogt H. Efficient Object Identificacition with Passive RFID Tags.; Proceedings of the International Conference on Pervasive Computing; Zurich, Switzerland. 22–26 August 2002; pp. 98–113. [Google Scholar]

[14] Klair D.K., Chin K.W., Raad R. A survey and tutorial of RFID anti-collision protocols. IEEE Commun. Surv. Tutor. 2010;12:400–421. [Google Scholar]

[15] M. Ewerbring, B. Gudmundson, G. Larsson, and P. Teder, CDMA with Interference Cancellation: A Technique for High Capacity Wireless Systems, Proc. ICC 93, Geneva, Switzerland, 1993, pp. 1901-6.

[16] Tutorial Point – Simply Easy Techniques https://www.tutorialspoint.com/cdma/cdma_techniques.htm

[17] Gopi Krishna G., Shubhankar Reddy Katta, Harshada Naik and Sarath Chandra Chaparla, RFID Tag Counting.

[18] Nikola Cmiljanic, Hugo Landaluce and Asier Perallos, A comparision of RFID anti-collision protocols for tag identification. Applied Sciences, 1 Aug 2018.