Improvement to the Anticollision Protocol Specification for 900MHz Class 0 Radio Frequency Identification Tag

Mamatha Nanjundaiah
Institute for Scientific Computing
Wayne State University,
Detroit, MI 48202
mamatha@wayne.edu

Vipin Chaudhary
Institute for Scientific Computing
Wayne State University,
Detroit, MI 48202
vipin@wayne.edu

Abstract

This paper proposes a modification to the existing anticollision protocol put forth in version 1.0 Protocol Specification for 900MHz Class 0 RFID Tag. The version 1.0 specification uses a binary tree approach to singulate one RF tag ID at a time. The proposed change reduces the overall read time of a given number of RFID tags by resetting to the appropriate node, for every consecutive read cycle. The present standard resets to the root node of the binary tree for every read cycle.

1. Introduction

The use of Radio Frequency Identification system has long had to deal with issues of cost, implementation technology, standards, security, privacy, etc., to finally be accepted worldwide. Apprehensions regarding the practicality of these systems are now a thing of the past. Research and development of this idea has been a continuous process to improve the capability, applicability and efficiency of the system.

One of the areas of research is the speed with which a given number of tags in the field of an RFID reader can be identified. EPC Global Inc. is the organization leading the development of industry-driven standards for the Electronic product code (EPC) to support the use of RFID. The organization has put forth a set of standards for the RFID industry through its Version 1.0 specifications [1]. Included in these specifications is the anticollision protocol for the 900MHz Class 0 Radio Frequency Identification Tag. It is to this protocol that we suggest a modification for increasing the read rate of tags.

Section 2 of this paper describes related work. Section 3 describes the anticollision protocol as given in the specifications of version 1.0. Section 4 goes on to explain the suggested modification to the above-mentioned protocol in order to improve its multi-tag read rate. Section 5 describes the simulation that was used to demonstrate the improved performance, and Section 6 discusses the results.

2. Related work

Prior work on anticollsion algorithms for RFID tags can be found in [2-7]. Hernandez *et. al* [2] discuss a technique where each tag sends out its ID data continuously with a pause between two consecutive transmissions, where the pause is independent for each tag. Here the probability of all tags being read increases with reading time.

Vogt [3] proposes a technique in which the reader broadcasts a request, containing an address range, which determines what data the tags should return, and what random number to be used as a seed by the tags in choosing a time slot. After the broadcast, N slots are provided for the tags to answer in. Here, optimum N is determined so as to maximize the throughput.

Jacomet *et. al.* [4] propose a technique similar to the binary protocol of [1]. The, tags respond with their next bit in the 1st or 2nd slot following the reader's command, depending on whether the bit's value is '0' or '1'. Hence, there will never be a conflict in response from tags.

Law *et. al.* [5] describe a query tree protocol that consists of rounds of queries and responses. In each round, if there is more than one tag that has the same prefix requested by the reader, then the reader appends a 0 and 1 to the same prefix and continues the queries. When a tag's ID matches the prefix uniquely, it is identified.

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

Zhou *et. al.* [6] compare the protocols given in [1] and [5] and suggest an improvement to the algorithm in [5] by way of cutting short the responses of tags that conflict.

3. Class 0 anticollision protocol

Figure 2 shows the tag protocol state diagram as given in the version1.0 Class 0 specifications.

Here we deal with tags that have individual binary IDs of either 64 bits or 96 bits length and respond to the reader's queries through backscatter modulation. Upon power up the tags enter the "Dormant state". They then wait for a "Reset signal", receiving which they transit to the "Calibration state", where they receive the calibration bits that leads them into the "Global Command Start state". If the tags, at any point of time, receive invalid signals, they transit to the states shown in the state transition diagram of Figure 2.

The tags then proceed to the "Tree Traversal state" upon receiving a bit "0" provided their ID Flag is cleared. The moment they come into this state the tags start shifting out their MSBs to the backscatter modulator. The reader looks at the response from the tags and depending on whether it receives a clear '1' bit or '0' bit it retransmits it. If there is a conflict of bits, depending on how it is programmed, the reader decides to pick one of the bits to transmit

If the next bit received is the same as the one the tags shifted out previously, they send the next bit of the ID. If not, they move to the "Traversal Mute state". This proceeds till the last bit position when a single tag's ID is identified. A data 'Null' transmission from the reader at this point brings all the tags in the Traversal Mute state to the "Tree Start state" and the identified tag to the "Singulated Command Start state". A'0' bit now puts the tags in Tree Start state back in the Tree Traversal state and the identified tag to the dormant state after setting its ID Flag.

Here we see that the protocol can be interpreted as a tree (illustrated in Figure 1), which is scanned from the root to leaf to identify the EPC of a single tag. The tree needs to be scanned this way for every tag present in the reader's field.

4. Modified Protocol

If there are many tags within the field of the reader that have values of their IDs such that, out of 64 or 96 bits, the first few MSBs have the same pattern, the time required to read all the tags can be reduced. Instead of always going to the root of the tree, the Tree Traversal cycle can begin with the bit position from where the last difference in the tag IDs occurred.

For example, in Figure 2, after the product 1("10001") is identified, instead of resetting to the root (level 0), the tags can be reset to node 4. The reader knows that tags (in conflict) having that same bit pattern till level 2 are sending their level 3 bits. Again, in this case, there is a conflict at level 4. So once product 2("10100") is identified, tags can reset to node 5 to recognize product 3("10101") and so on.

This protocol does increase the complexity of the tag by including a conflict-counter and a conflict-bit-pointer. While in the Tree Traversal state, when a tag conflicts at a certain bit, it records the bit position in the conflict-bit-pointer, increments its conflict-counter by one and moves to the Traversal Mute State. While in this state, it increments the conflict-counter every time it detects a conflict between the rest of the tags in the Tree Traversal state and decrements the counter for every successful read. When the tag receives a 'Null' bit after another tag has been successfully read, it transits to the Tree Start state. A '0' bit now will put the tag in Tree Traversal state or Traversal Mute state depending on whether the conflict-count is zero or not.

When the conflict-count reaches zero, the tag starts the cycle by shifting out the bit in the conflict-bit-pointer position. If there is a conflict again during the cycle, the tag repeats the above procedure, until it gets recognized. This modification has been shown in Fig.2 as a dashed box.

5. Simulation Model

The Class 0 protocol and our suggested modification to the protocol have been simulate using CSIM18. "Mailboxes" have been used to communicate the messages between reader and tags. The time for communication of bits '0','1', 'Null' and "Reset" and "Calibration" bits were calculated as per the specifications given in [1], and they have been given as "Hold" times. Ideal conditions have been assumed so that there is no near-far effect, or errors in the communication between reader and tags.

The number of tags has been varied from 5 to 85 in steps of 5. For each count of tags, 32-bit tag IDs have been generated randomly using "Uniform Distribution", "Triangular Distribution", "Beta Distribution" and "Geometric Distribution", over the entire range of 0 to (2^32-1). The results of these simulations have been illustrated in Figure 3.

6. Results

Figure 3 shows the result of simulation for the case of tag IDs generated randomly using Uniform Distribution. An overall average improvement over the binary protocol [1] of 11.44% was noted. Figure 4 shows the result of the case of Triangular Distribution. Here mode was taken as mid-point of the given range. An average improvement of 11.09% was obtained. Figure 5 gives the result for the Beta Distribution case. The (shape) parameters were taken as 3.0 each. This case gave an improvement of 11.08%. Figure 6 gives the result for the Geometric Distribution case. Here the probability of success parameter was taken to be 0.00000001. This case gave an improvement of 20.44%. The average improvement curve of Figure 7 was obtained by taking the average of the difference in read times for the 2 protocols, for given number of tags. Figure 8 gives the same result in terms of percentage.

7. Conclusion

From the results of the simulation we have obtained an overall improvement of 13.52%. From the graph showing the "Average percentage improvement" it is clear that the impact of our proposed scheme increases with increase in number of tags. This can mean huge time savings specially in areas of RFID application like warehouse inventory management, where large groups of goods can have similar initial bit patterns. We are looking at the impact of near-far effect on our proposed changes.

8. References

[1] "Draft Protocol Specification for a 900MHz Class 0 Radio Frequency Identification Tag", MIT Auto-ID Center, 23rd Feb 2003, www.epcglobalinc.org

[2] P. Hernandez, J. D. Sandoval, F.Puente and F.Perez, "Mathematical Model for a Multiread Anticollision Protocol", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Aug 2001, pp. 647-650

[3] Harald Vogt, "Multiple Object Identification with Passive RFID Tags", *IEEE International Conference on Systems, Man and Cybernetics,* Oct 2002. Volume 3: pp. 6-9

[4] Marcel Jacomet, Adrian Ehrsam, Urs Gehrig,"Contactless Identification Device With Anticollision Algorithm", 1999. http://www.microlab.ch/pdf/Papers/cscc99.pdf

[5] Ching Law, Kayi Lee and Kai-Yeung Siu, "Efficient Memoryless Protocol for Tag Identification", *Proceedings of the 4th ACM Iinternational workshop on Discrete algorithms and methods for mobile computing and communications*, Aug 2000, pp. 75-84.

[6] Feng Zhou, Dawei Jin, Chenling Huang and Min Hao, "Optimize the Power Consumption of Passive Electronic Tags for Anti-collision Schemes", *ASIC Proceedings of 5th International Conference,* Oct 2003. Volume 2, pp. 1213-1217.

[7] Don R. Hush and Cliff Wood, "Analysis of Tree Algorithms for RFID Arbitration", *IEEE International Symposium on Information Theory,* Aug 1998, pp. 107.

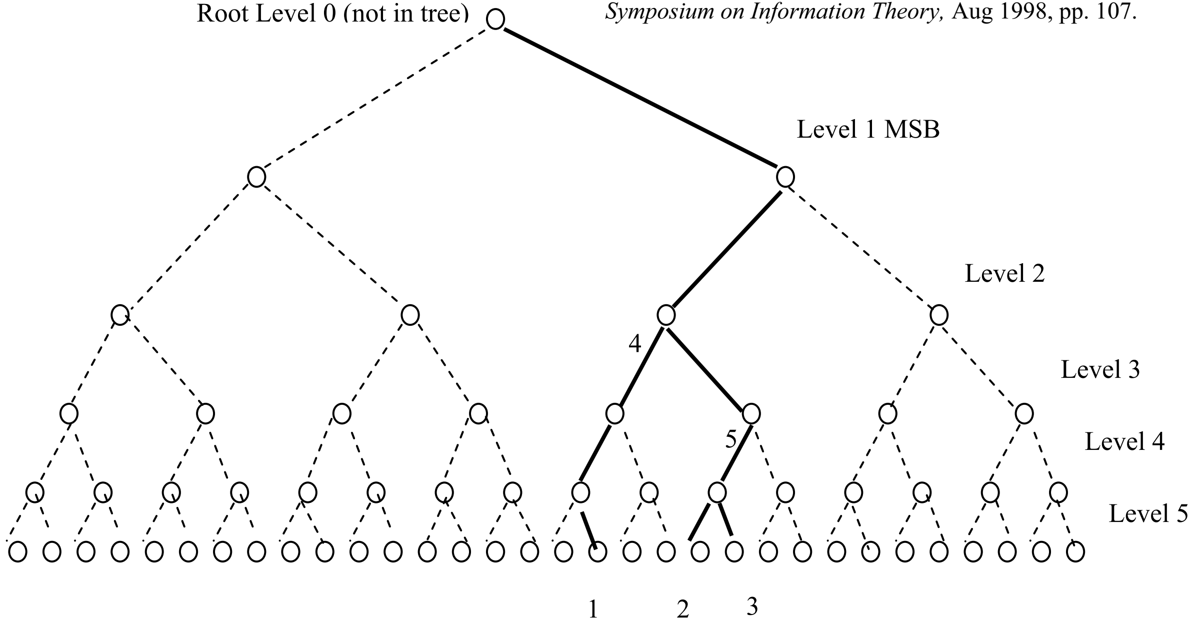

Figure 1: Representation of an EPC as a tree

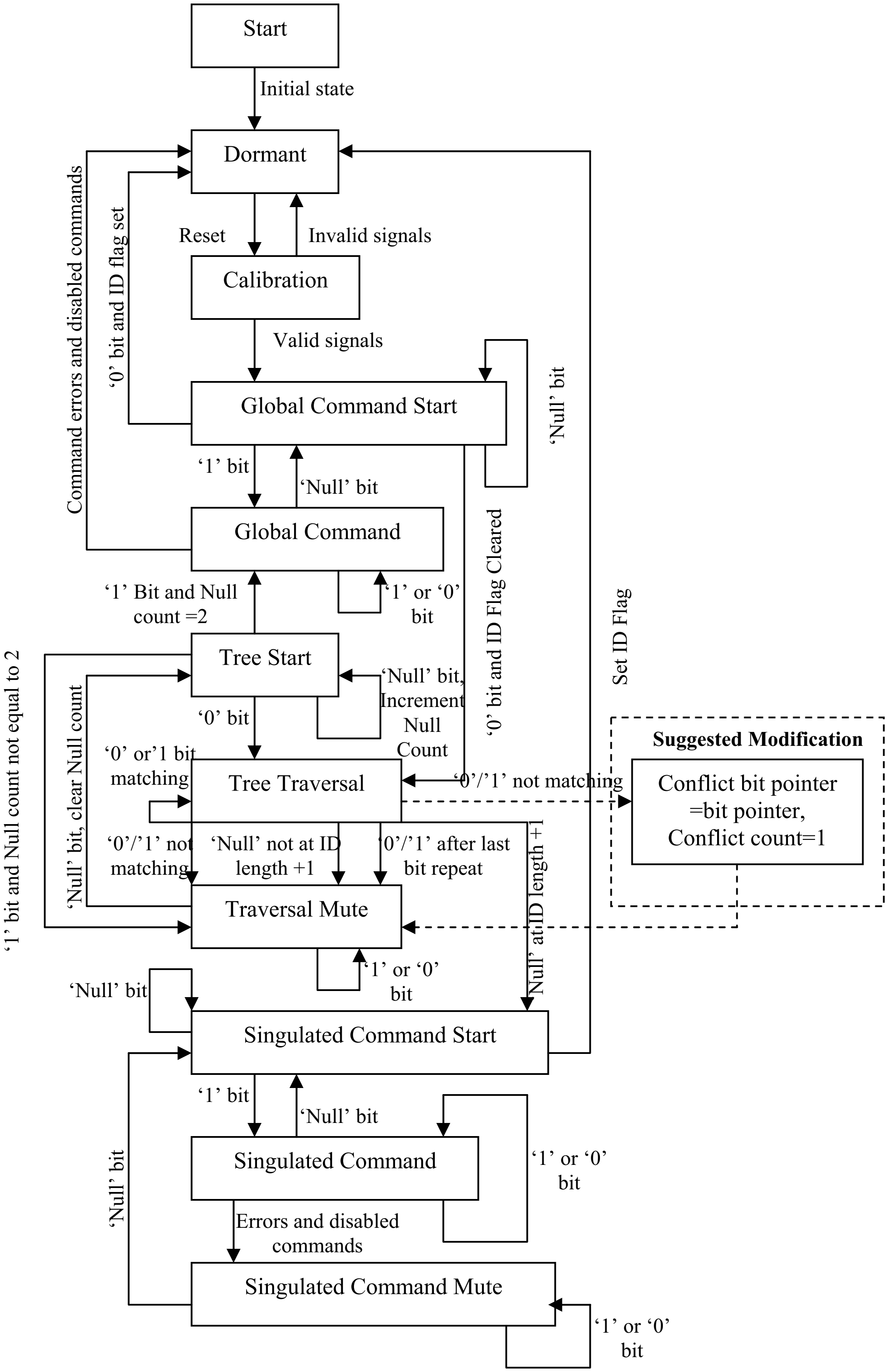

Figure 2: Class 0 Protocol State Diagram

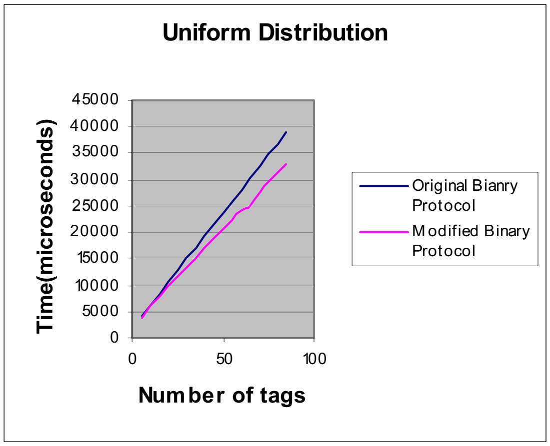

Figure 3: Simulation result for case of Tag IDs generated by Uniform Distribution

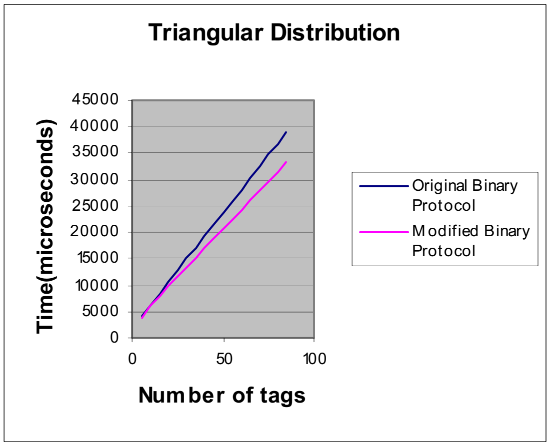

Figure 4: Simulation result for case of Tag IDs generated by Triangular Distribution

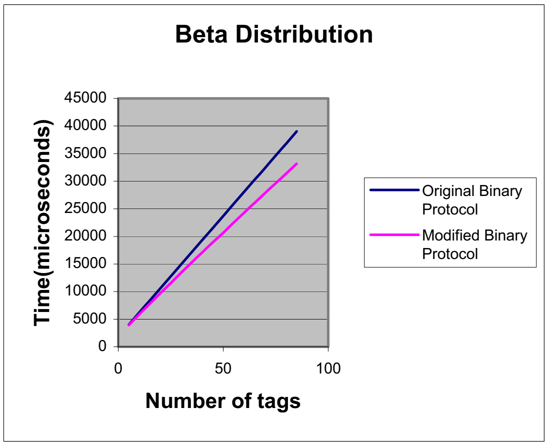

Figure 5: Simulation result for case of Tag IDs generated by Beta Distribution

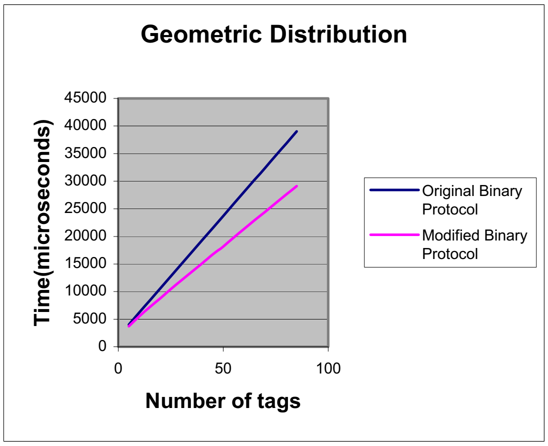

Figure 6: Simulation result for case of Tag IDs generated by Geometric Distribution

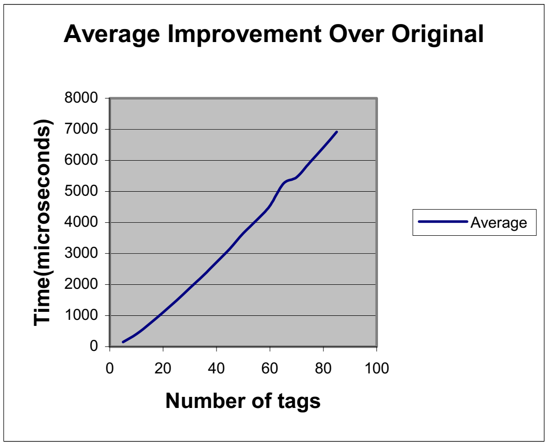

Figure 7: Average Improvement in Time

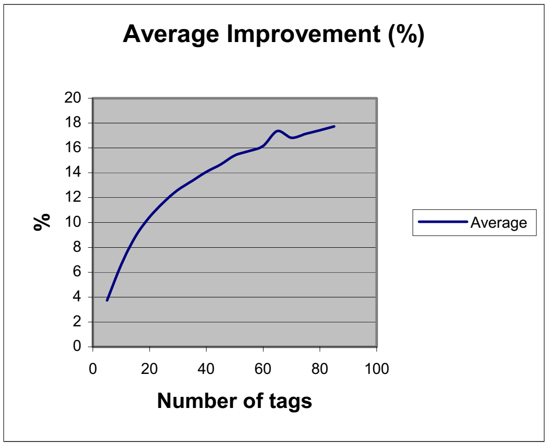

Figure 8: Average Percentage Improvement