Fast and Simple Anti-collision Protocol based on Up-down Counter and one Bit Reader Response

Usama S. Mohammed and Mostafa Salah

Department of Electrical Engineering, Assiut University, Assiut 71516, Egypt

Abstract. This paper describes a simple fast protocol of counter based anti-collision approaches. The tag processing is depending on one depth counter and conflict pointer to determine its replying order and the current replying bit. The main advantage of the proposed protocol is the simple one bit reader response in the identification process. The one bit reader response will provide the tags all information about the new state of the current depth position of replying. The tags will transmit its IDs without reader interruption until the reader detects the collision state or the identification state. Based on this idea, the identification process will be faster than the traditional counter based anti collision protocols and the overhead information will be reduced. Performed computer simulations have shown that the collision recovery scheme is very fast and simple. The simulation results shows that the proposed technique outperforms most of the recent techniques in most cases.

1 Introduction

The RFID systems consist of networked electromagnetic readers and tags, where the readers try to identify the tags as quickly as possible via wireless communications. However, since the readers or the tags communicate over the shared wireless channel, the collision problem occurs in signal transmission of the readers or the tags, which leads to slow identification. Thus, it is a key issue to develop an efficient anti-collision protocol reducing collisions so as to identify all tags in the interrogation zone. Collisions are divided into reader collisions and tag collisions [8]. Reader collision problems arise when multiple readers are simultaneously used. The other, most important, collision problem (approached in this paper) is the tag collision that occurs when several tags try to answer to a reader query at the same time.
Messages would collide on the communication channel and thus cannot be interrupted by the reader. Resolving collisions has been a consistent research subject in wireless communication, included RFID systems. Passive tags take its power from reader RF signal, and use load modulation by reflecting energy from the reader for setting up a communication to the reader. Tags have to be very simple in order to be as cheap as possible. The classical multiple access anti-collision schemes cannot be directly applied to the tag identification problem due to various constraints, which make this problem unique. Anti-collision protocols designed for tag collision problem have some constrains due to the nature of passive tag:

S. Mohammed U. and Salah M.
Fast and Simple Anti-collision Protocol Based on Up-Down Counter and One Bit Reader Response.
DOI: 10.5220/0003017300970107
In *Proceedings of the 4th International Workshop on RFID Technology - Concepts, Applications, Challenges (ICEIS 2010)*, page
ISBN: 978-989-8425-11-9

1- Lack of internal power source in the passive tags. This requires the tag reader to power-up these tags whenever it needs to communicate with them.
2- Total number of tags is unknown. 3- Tags cannot communicate with each other. Passive tags cannot figure out neighboring tags or detect collisions. Hence collision resolution needs to be done at the reader. 4- Limited memory and computational capabilities at the tag.
These constrains are due to the requirement to keep the tags as cheap as Possible. Thus the resolution protocol must be simple and incur minimum overhead from the tag's perspective.

2 Related Work

In RFID system, there are two approaches of tag collision resolution scheme: (1) Probabilistic algorithm which is based on ALOHA. (2)Deterministic algorithm (tree based protocols) which detects collided bits and splits disjoint subsets of tags.
In generally, Aloha based protocols cannot perfectly prevent tag collisions because of the probabilistic procedure. Moreover, they have the serious problem, called "tag starvation", that a tag may not be identified [9]. Meanwhile, the tree-based protocols such as the tree based protocols (query tree & binary tree) do not cause tag starvation occurring in the aloha-based protocols although they require relatively long time to identify all of tags. In that paper, the binary tree based protocols will be modified based on up-down depth counter.
The reader in query tree (QT) protocols sends a query containing a prefix having length of 1 to n bits. The tags whose prefixes match with the bits sent by the reader, replies back with their tag ID. The reader asks the tags to answer if their ID matches a given prefix. If there is a collision, the reader queries for one bit longer prefix until no collision occurs. The only computation required for each tag is to match its ID against the binary string in the query. [1] There are different schemes of the basic query tree protocols as in [2], [3], [4], [5], [6] for reducing the exchanged overhead between the reader and tags, and to have shorter identification time.
Deterministic resolution of collisions based on binary tree algorithms is done by successively splitting tags into two subsets, directs tags to either remain in active mode, or go temporarily inactive. Only one tag will be left sending its data at last. In this work, a simple and fast counter based binary tree anti-collision protocol is proposed. There are different schemes of counter based protocols as in [7], [8], [9], [10], [11] and [12].
In [7] the tag uses its internal counter to determine when it changes the state from Quiet state to active state. But, the tags in active mode need to know the last collision position. Hence, the reader must send query bit of the last collision bit position after each identification to inform the tags with the last stop bit position.
In [8], the basic search criterion is the depth-first search (DFS) algorithm. A height-oriented method is used as encoded-binary number corresponding to the height of the vertex at which the tag collision most recently occurred. This process reduces the redundant of the transmission bit length. Each tag has a pointer, this pointer will move toward a lower bit with the ongoing of inquiring, and a reader has the function

of recording the position of the vertex at which the tag collision occurred. Figure 1 shows the reader command frame, the flow diagram of searching tags and the bit stream between the reader and four tags (0001,0011,1000,1100). Moreover, Figure 2 shows the state diagram of that protocol presented in [8].
Large overhead in this protocol is due to: (1) The need to send the height of the most recently occurred collision bit position after each identification to inform the inactive tags about the stopping bit position. (2) Reader command frame bit is relatively high.

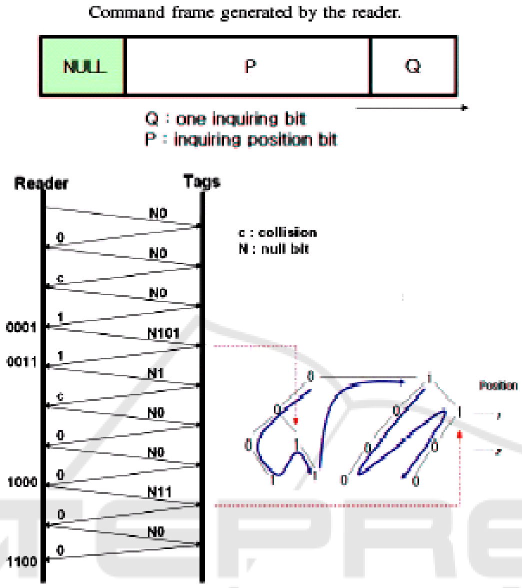

Fig. 1. Bit stream between a reader and tags and the flow diagram of searching tags. [8].

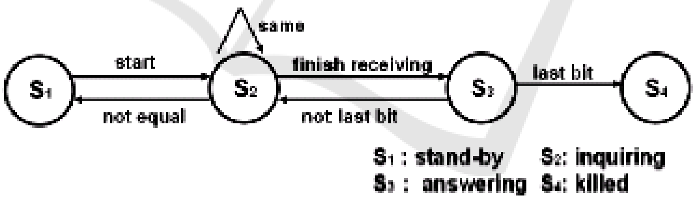

Fig. 2. State diagram. [8].

In [9] and [10], adaptive binary splitting (ABS) uses counter to reach the goal of anti-collision, but the splitting of sets depends on the random binary number {0, 1}. So, it cannot achieve the best splitting result. The probability of occurrence of 1 or 0 is not 50%. At any moment, there won't be any splitting result, and may cause the next timeslot to be idle timeslot or collided timeslot. Consumption of timeslots and longer timeslot are the main drawback of the ABS protocol.

In [11], the main assumption is the reader ability of truncating unnecessary data bits to reduce the receiving time in the collision condition. It means that, for the reader, it is not necessary to receive any data after receiving the second collided bit.

Moreover, the reader will detect the type of the timeslot. Manchester code is used in binary search algorithm, to find the position of the collision timeslot. It defines three kinds of timeslots by three bit operation code: collision, identified and OBCT (One Bit Collision Time slot). In this approach, transmitting a part of ID starting from collision bit to the end of the ID is used in the identification process. So the length of the time slot will be decreased. The main drawback of this approach is the long reader feedback message. Moreover, in the collision state, it must feedback the position of the first collided bit.

Recently in [12], a new approach for anti-collision algorithm is introduced. This approach is based on parallel binary splitting (PBS) technique to follow new identification path through the binary tree. The advantages of this scheme are the low implementation complexity and the lower number of transferred bits between the reader and the tags in the identification process. It has a parallel tree-scan based on self modification of the tags relative order. Although, the PBS protocol minimizes the dialog between the reader and tags to only one bit tag response followed by one bit reader reply, it provides overhead information of the reader report to provide the state of collision or no collision. The number of exchanged bits is equal to twice the number of the binary tree nodes of the existing tags except the leaves nodes.

A new idea in [13] is introduced to reduce the probability of collision efficiently and to make fast identification of multiple passive tags in a timeslot. It reduces the length of the time slot by truncating unnecessary data bits to minimize the receiving time. The reader does not need to receive any data after receiving the first collided bit. It classifies the tags into sets according to the number of ones "1" in each tag-ID. Then it can identify tags set by set. In this algorithm uses four types of timeslots are used. These time slots are: collision, readable, M-readable (Multiple-readable), and idle. A feedback message is send by the reader to inform the tags about the type of a timeslot. Based on this message, all tags can make a suitable response for the next timeslot. When collision occurs, the reader will inform all tags about the collided bit. Feedbacks are just like instructions and include operating code and some other information. The operating code in 3 bits is used. It improves the collision resolution, reduces the length of the time slot and makes upper bound of the total time slots is needed for identification. So, it is a long reader instructions and the reader must inform the tag the position of collided bit in the collision code. In the next section, we will show how to overcome the main drawbacks in the previous work using simple one bit reader response. The bit position of last collision bit is not requested. The reader needs to respond only one bit in the collision or identification state.

3 Proposed Protocol

Assuming that inside each tag there is a depth counter and collision pointer. The depth counter (D) will measure how far the tag position from the current replying tags. The collision pointer (P) will keep tracking the marked bit that will be transmitted when being in the active replying state. The tags will transmit the ID bits without reader interruption until the reader detects collision or identification.

3.1 Reader Operation

The reader starts the communication session by transmitting the continuous RF signal to power the passive tags. Then, all tags will be in the active state and the ID, bit by bit, will be transmitted from tags in active mode. The reader then receives the tags responses until the collision state (at collision node) or finishing the identification path (at leaf node). I.e. the tags will transmit its IDs without reader interruption until the reader detects collision or identification. The reader one bit response (1 for collision and 0 for identification) will be sent to inform the tags how to change there current replying depth. If the reader detects a collision state at the end of any path in the binary tree, it will be treated as two tag identification. Hence, it sends the identification response. The reader will return to the nearest waiting tags whose depth counter becoming zero.

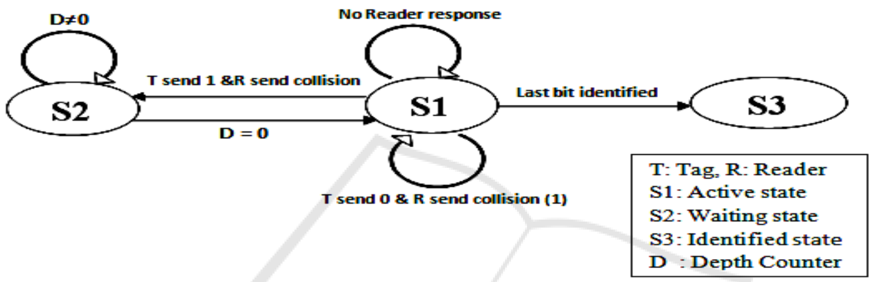

Fig. 3. State diagram of the proposed protocol.

3.2 Tag Operation

All tags will start in the active state as response to the reader command. The current active tags will continue replying its ID bit by bit until completing identification or receiving a collision notification from the reader. In the case of receiving the collision notification, there are two possibilities: 1) the active tags replying 0 will continue in the active state without affecting the depth counter, 2) the active tags replying 1 will stop transmission and increment its depth counter. Then the tags will be in the waiting state. It stays silent and the reader will continue in the path of the active tags. It is important to note that, the depth counter (D) for the tags in the waiting state will be incremented at collision acknowledgment, and will be decremented at identification acknowledgment. Also, the depth counter for the waiting tags will be decremented by the reader identification response until zero count (D=0), then the state will be changed to the active state. There is no need to tell the tag any information about the position of starting node. The information about the last bit shifted out is determined by the collision pointer. The tag will be in the identified state when it has sent all its ID bits. Figure 3 shows the State diagram of the proposed protocol.

4 Performance Analysis

In this section, the performance of the proposed algorithm will be discussed. Figure 4 shows the diagram of four tags in binary tree, as an example, **{A, B, C, D} = {0000, 0110, 1110, 1111}** to be identified with each node contains its replying order according to the Depth First Search (DFS) path.
By looking to that figure, the reader needs to hear the one bit reply of the interrogated tags at each node. Tags and the reader agree to follow the interrogation order of these node according to the proposed protocol. Figure 5 shows the bit-exchanged stream between the reader and tags during the identification session.
As mentioned, the reader responds only to inform the tags with the branching moment in the collision state or the return action to the identification state.

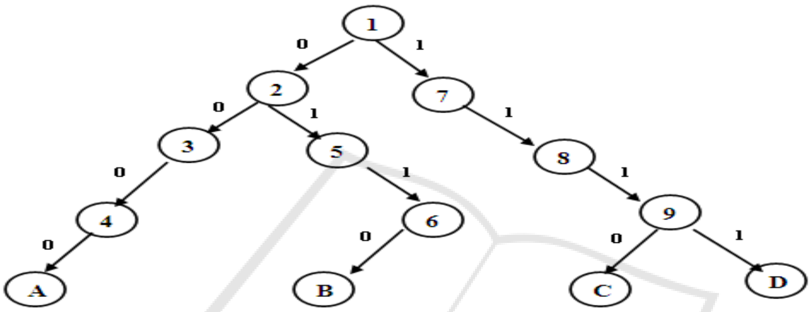

Fig. 4. The diagram of four tags binary tree.

Reader reconstructs the binary tree by scanning the tree nodes in the shown order in Figure 6. This Figure describes in details the process of node exploration under the control of its depth counter. The contents of depth counter are shown also in this Figure after each step.

In this example, the performance analysis of the proposed algorithm is proposed in a comparison form with the recent algorithms as follows:
For the number of tree nodes (except the leaves node) = 9 nodes.

- In the EAA algorithm [11], the total number of feedback bits and the reader response is 19 and 11; respectively. It uses 30 bits to identify the tree nodes in our example. (19+11=30).
- In the PBS algorithm [12], it consumes one bit for each node (tag response), and one bit for reader to report the state (collision or no collision). The number of transmitted bits by the tag = the number of transmitted bits by the reader = 9 bit. Then, the overall transferred bit =18 bits.
- In the proposed algorithm, it consumes 9 bit as tags response and 5 bit as reader response. The overall transferred bit between the reader and tags = 14 bits. This improvement will be more sensed by increasing the number of identified tags and the number of bits per tag.

If we consider the reader responses as the overhead information in the process of tag identification, the overhead in the proposed protocol can be considered two bits per one tag as maximum. Hence, in the proposed protocol the Max. Exchanged bits =

2*number of existing tags + Number of the binary tree nodes except leaves.
In general, if the tag has *n* bit length, then the overall time needed for one tag identification must be less than the time needed for the transmission of *(n+2)* bits, which is the upper bound of the transmitted bits per tag.

However, the upper bound of the algorithm in [13] is computed relative to the number of time slots (it has a long timeslot). It has four type time slots with three bits instruction code, besides sending the position of the collided bit along with the collision code.

Figure 7 and table 1 show a comparison between Dynamic Bit Arbitration (DBA) in [6], Parallel Binary splitting (PBS) in [12], and the proposed algorithm by measuring the total transferred bits between the reader and tags (under 32 bit ID long).

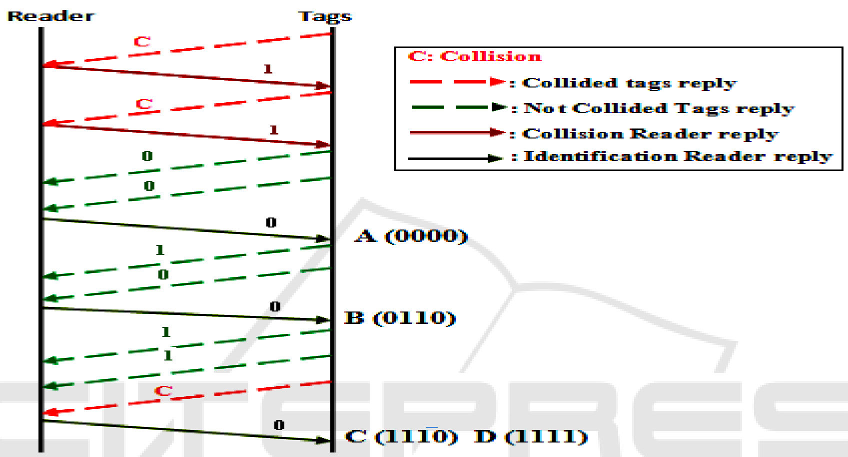

Fig. 5. Transmitted bit stream between the reader and tags.

Table 1. Estimated number of exchanged bits.

no. of tags	Exchanged bits (Random IDS with 32 bit tag ID long)		
	DBA [6]	PBS [12]	Proposed algorithm
50	3024	2650	1425
100	5894	5098	2749
150	8638	7458	4029
200	11392	9664	5232
250	14060	12014	6507
300	16726	14328	7764
350	19332	16572	8986
400	21952	18800	10200
450	24478	20982	11391
500	27186	23138	12569

In the simulation results, to identify 250 tags, DBA transfers 14060 bit while PBS transfers 12014 bit. However, to identify the same number of tags using the proposed algorithm, only 6507 bit is required as the total exchanged bits, including 500 bits are the maximum reader overhead. Figure 8 shows the results of the identification time with respect to the number of tags when the proposed algorithm and four of recent algorithms are used with 40kb/s bit-rate and the ID length is 16 bits. In this Figure, for the BIBD technique [5], at least 450 ms is required to identify 300 tags and 2550 ms is required to identify the same number of tags by the traditional query tree algorithm. The dynamic query tree algorithm identifies the same number of tags in 1500 ms. However, the proposed protocol can identify the same number of tags within 60 ms.

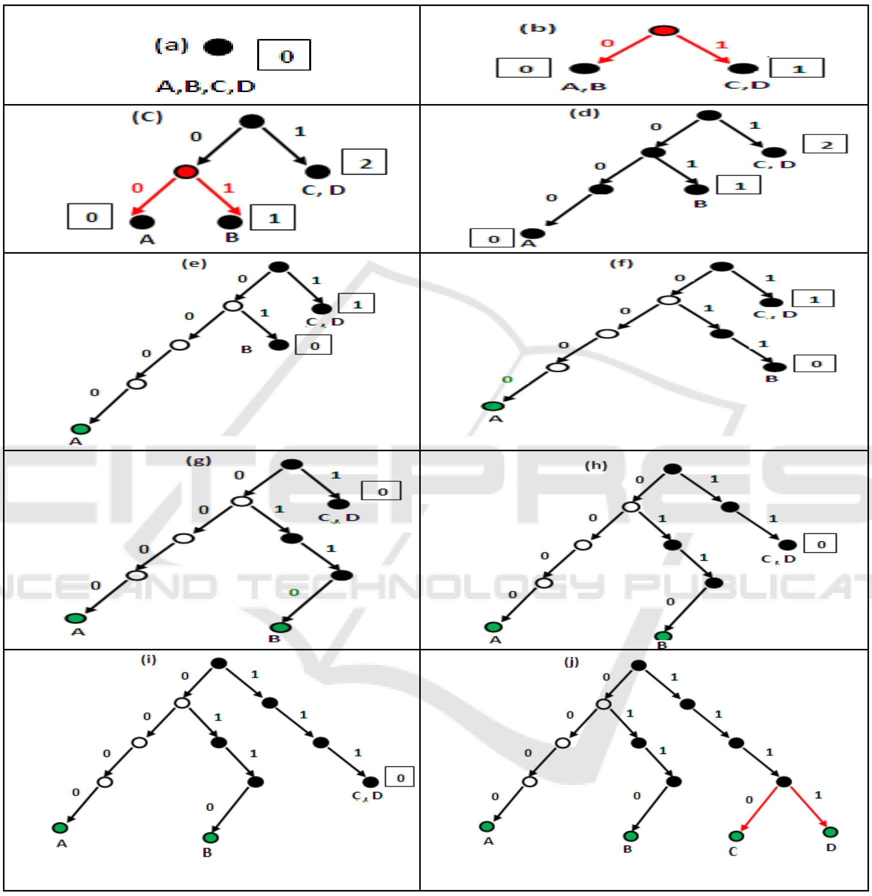

Fig. 6. The order of node exploration.

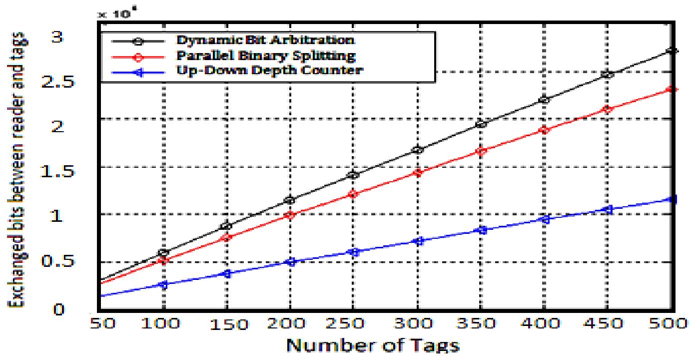

Fig. 7. Total transferred bits vs. the number of tags, for random IDs with 32 bit ID long.

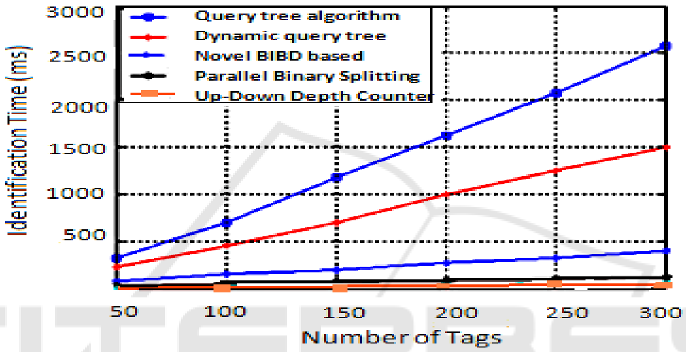

Fig.8. Identification Time of different number of tags with 40kb/s bit rate (ID length =16 bits).

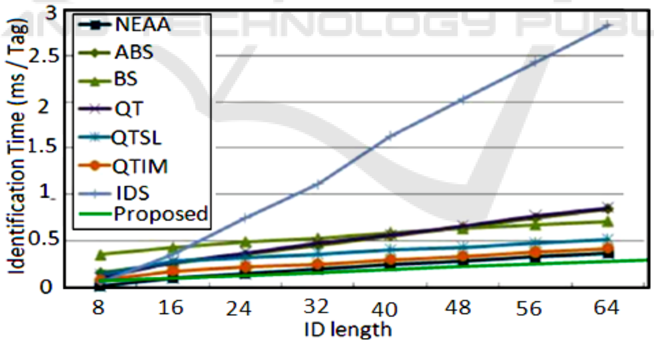

Fig. 9. Average identifying time of each algorithm.

Fig. 9 shows the results of identifying 256 tags with variable ID length from 8-bit to 64-bit. Assume that the time of transmitting one bit is 5 μs under bit rate of 200kb/s. Then, if the proposed protocol is used to identify 256 tags of 8 bit ID length (the 100% tag density of 8 bit ID length i.e.2^8=256 tags are exist in the interrogator's operating range), the reconstructed binary tree has 255 internal nodes (without including the tree leaves). It needs 1 bit tags reply at each node. There is a collision in each node provides 1 bit reader collision response and 128 bit reader identification. The total number of transferred bits is 638 bit. The total identification time =638* 5 µs= 3190 µs. It provides 0.0124 ms per one tag identification. However, the algorithm in [13] consumes 0.0168 ms to identify one tag.

In [13], an example of identifying seven tags A, B, C, D, E, F, and G in the interrogating zone is proposed. Their tag IDs are "0000", "0001", "0010", "0110", "1001", "1010", and "1110", respectively. The total number of transmitted bits between tags and reader are 23+19=42 bits. The total number of feedback bits and the total number of response bits are 23 and 19, respectively. [13]

However using the proposed protocol, only 13+11=24 bits are used to identify the same number of tags. As shown in Fig. 10, the constructed binary tree of the seven tags has 13 internal nodes. So, the tags will reply by 13 bits and the reader will send 5 bits for collision notification and 6 bits for identification notification.

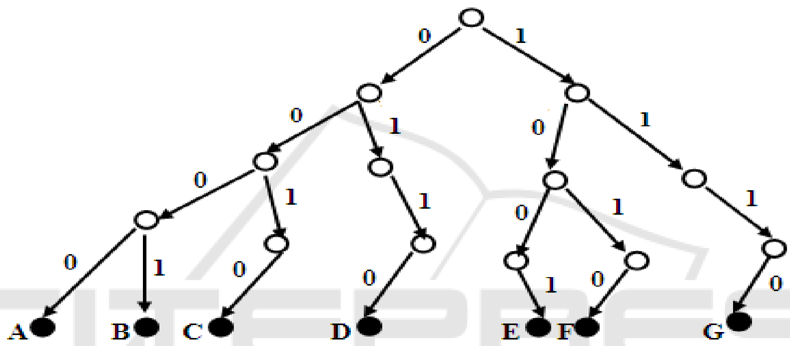

Fig. 10. The diagram of the seven tags binary tree.

5 Conclusions

This paper presents simple and fast anti-collision algorithm. It overcomes the main drawbacks of counter based anti-collision protocols. It depends on only one up-down counter controlled by simple logic and one bit reader response. It achieves great save in the reader overhead by using simple one bit response (1 for collision, 0 for identification). The tags will transmit its ID bit by bit without reader interruption until the reader detects collision or identification. The tag state will be changed from the waiting state to the active state without any information about the position of the restarting node. The simplicity of the proposed algorithm is due to the low cost of integration in the passive tag. Moreover, the proposed algorithm utilizes the identical prefixes IDs. It works with minimum overhead. The simulation results have shown that the collision recovery scheme is very fast and simple relative to the recent anti collision protocols in literatures.

References

1. A. Sahoo, S. Iyer and N. Bhandari, “Improving RFID System to Read Tags Efficiently”, Kanwal Rekhi School of Information Technology, Indian Institute of Technology Bombay, approved for the degree of Master, June, 2006
2. Jung-Sik Cho, Jea-Dong Shin, Sung K. Kim, RFID Tag Anti-Collision Protocol: Query Tree with Reversed IDs, Chung-Ang University, Seoul, Korea, , the 10th International Conference on Advanced Communication Technology, Volume: 1, Page(s): 225 – 230, Feb. 17-20 ICACT 2008
3. Ji H. Choi, Dongwook Lee, and Hyuckjae Lee, “Query Tree-Based Reservation for Efficient RFID Tag Anti-Collision”, IEEE Communications Letters, VOL. 11, NO. 1, Page(s): 85 - 87 January 2007
4. Okkyeong Bang, J.H.Choi, D.W. Lee ,H.J.Lee, “Efficient Novel Anti-collision Protocols for Passive RFID Tags,” Auto-ID Labs White Paper WP-HARDWARE-050 March 2009. www.autoidlabs.org
5. Shiyu Li and Quanyuan Feng, A Novel Anti-collision Algorithm in RFID System, In Electromagnetics Research Symposium, Cambridge, USA, 207-210, (2008).
6. Ziming Guo, and Binjie Hu, “A Dynamic Bit Arbitration Anti-Collision Algorithm for RFID System”, IEEE International Workshop on Anti-counterfeiting, Security, Identification, 457 – 460 (2007).
7. Jun-Bo Guo, and Zhen-Hua Ding, ID-binary tree stack anti-collision algorithm for RFID, Proceedings of the 11th IEEE Symposium on Computers and Communications, Page(s): 207 - 212, 2006, ISCC '06.
8. Sung Hyun Kim, PooGyeon Park, An Efficient Tree-Based Tag Anti-Collision Protocol for RFID Systems, IEEE Communications Letters, vol. 11, No. 5, 449-451 (2007).
9. Jihoon Myung, Wonjun Lee, Jaideep Srivastava, “Adaptive binary splitting for efficient RFID tag anticollision,” Communications Letters, IEEE, Vol. 10, Issue 3, pp.144-146, Mar. 2006.
10. Jihoon Myung, Wonjun Lee, Jaideep Srivastava, and Timothy K. Shih, “Tag-splitting: adaptive collision arbitration protocols for RFID tag identification, IEEE Transactions on Parallel and Distributed Systems, Volume: 18 , Issue: 6 , Page(s): 763 – 775, 2007.
11. Wei-Chih Chen, Shi-Jinn Horng, and Pingzhi Fan, An Enhanced Anti- collision Algorithm in RFID Based on Counter and Stack, Proceedings of Second International Conference on Systems and Networks Communications (ICSNC 2007), Issue 25-31, (2007) 21 - 21.
12. U. S. Mohammed and M. Salah, Parallel Binary Tree Splitting Protocol for Tag Anticollision in RFID Systems, Proceeding of the 4th IEEE international workshop in Design and Test (IDT09), Riyadh, Saudia Arabia, Nov. 2009.
13. Yuan-Hsin Chen, Shi-Jinn Horng, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, Wei-Chih Chen, Yi Pan, and Terano Takao, ” A Novel Anti-Collision Algorithm in RFID Systems for Identifying Passive Tags”, IEEE Transactions on industrial informatics, VOL. 6, NO. 1, February 2010.