Tag Starvation and Tag Collisions in RFID System- A Solution

Prof. Nithya S [#1], Prof.Asha Jerlin M [*2], Prof. Swarna Priya RM [#3]
[#]School of Information Technology and Engineering, VIT University Vellore, Tamil Nadu, India
[1] nithya.s@vit.ac.in
[3] swarnapriya.rm@vit.ac.in
[*] School of Information Technology and Engineering, VIT University Vellore, Tamil Nadu, India
[2] ashajerlin.m@vit.ac.in

***Abstract*—Radio Frequency Identification (RFID) is entirely based on tag identification. Reader identifies a tag by its ID. Collisions occur frequently while reading the passive tags and reader fails to identify the tags sometimes. This will lead to tag-starvation problem. Therefore, an algorithm is developed to address both collisions and tag-starvation problem. This paper uses Query Tree protocol and ALOHA to develop the algorithm. Query Tree protocol is used to address tag-starvation problem and ALOHA to reduce the collisions of passive tags.**

Keywords—RFID; ALOHA; Query Tree Protocol.

I. INTRODUCTION

RFID is an automatic identification system which includes the communication between reader and tags over a wireless channel. Each tag has its own ID with a storage capacity in order to store some related data. Reader must be capable of reading these tags as quickly as possible. Reader could only read any tag when the tag is in its range. RFID became a wide application in daily life due its automatic functionalities. Currently in several systems like libraries, transportation, cattle maintenance, etc.

Two types of tags are in use; *active* and *passive*. Active tags are assisted with a battery for power supply and are energized always. Its life time is less when compared to passive.

Active tags have a limited life time where as passive tags have a long life. Passive tags power supply is given by reader and is energized when reader is activated.

This paper deals with problems faced when using passive tags. Collision is the common problem in using passive tags. Reader must read several tags at a time and this leads to collisions. These collisions might cause in reader-tag communication or in tag-reader communication. Due to these collisions there is a problem that reader sometimes drops the tags without reading them. This problem is called as *tag-starvation* problem. This leads to information loss and wrong entries. As a result the efficiency and reliability are both decreased.

In order to deal with the problem discussed above an algorithm based on ALOHA and Tree Protocol is developed. The combination of two different methods gives an efficient solution. ALOHA usage reduces the collisions. Using ALOHA alone cannot remove the tag-starvation problem. Using ALOHA with Tree Protocol helps to remove the tag-starvation problem.

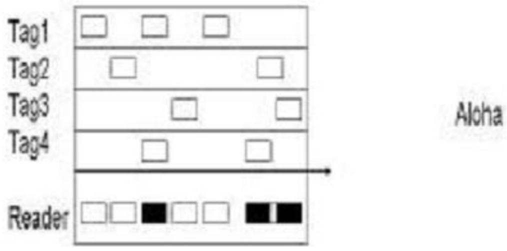
II. RELATED WORKS

A lot of protocols have been developed to eliminate the problem of collisions caused by the passive tags. Many algorithms with frame-slotted ALOHA has been developed to reduce the collisions. Several Query Tree protocols have also been developed to reduce the collisions.

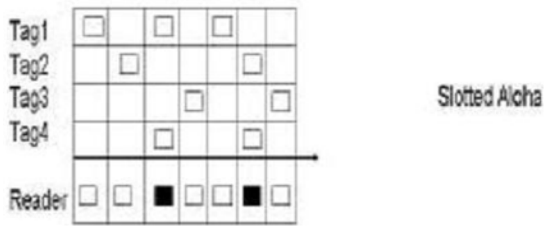
Tag identification accuracy is the vital point to be kept in mind [4]. All the tags must be identified else this leads to tag-starvation problem. To address this problem an algorithm using Query Tree (QT) protocol has been developed [3]. Not only query tree but also on the basis of binary tree protocols.

1. ALOHA based algorithms

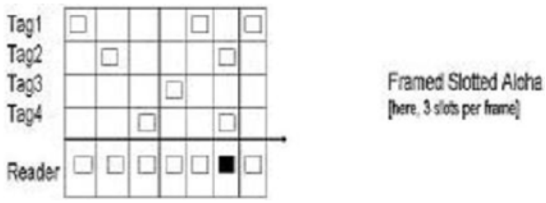
In ALOHA based algorithms, three types of ALOHA are possible. Aloha, slotted aloha and frame slotted aloha. Tags on identifying collisions are occurred try to select a random number and then transmit the messages according to their time slots. Then collisions occur if two tags select same time slot and try to transmit the messages. The collision reduction rate is maximum 18.4%. Tags undergo the same cycle iterations whenever collisions occur.



The slotted ALOHA maximizes the output and reduces the collisions at a rate double the ALOHA. Each tag selects a time slot and waits to start. When its time slot begins, tag starts to transmit and stops at the end of timeslot. This reduced the collision rate to 36.8%. Here packets collide completely or doesn t collide at all. The percentages depend entirely on the channel utilization. Slotted ALOHA utilizes double the channel used by ALOHA.



Frame slotted ALOHA on the other hand reduces the collisions by effectively utilizing the channel. Tags here, selects a tile slot like in slotted ALOHA and then a frame is allotted to some slots. Within a selected frame a tag can transmit only once. The frames number depends on number of tags being used. And the number of frames and slots decide the utilization of channel.



Frame slotted ALOHA reduce the collisions to a greater extent but can t eliminate the tag-starvation problem. An algorithm for eliminating tag-starvation problem is developed using QT protocol [3].

2. Query Tree Protocol

This algorithm is entirely for reader. Reader sends a query including the binary code. Tags on receiving the query checks if its tag ID is equal to the binary code. If so then tag responds to reader. If two or more tags respond to reader then a collision occur. Then reader adds a binary digit (0/1) to the query and stores in a queue. Then by selecting one query at a time and then repeats the cycle. If only one tag is responding to the query of reader or if no tag is responding to query then reader finds no collision and stores the query in another query.

QT algorithm also includes the collisions and is a slow process of identifying the tags. In order to reduce the collision rate and improve the identification accuracy and also to eliminate the tag-starvation problem an efficient algorithm is developed which includes both the ALOHA and QT concepts.

III. PROPOSED WORK

The algorithm being discussed is based on ALOHA [1] and Query Tree algorithm [3]. Using the algorithm from query tree algorithm, a small modification is made to increase the efficiency of the algorithm. It has two operations for tags and reader separately. Tag operation uses the ALOHA concept.

As the first step, the reader sends a query consisting of binary bit. Tags on receiving the query process it and responds if the query is equal to its ID. Then on receiving responses reader finds out if any collision occurs. If any collision takes place, reader sends a „gap-pulse indicating collision to tags and then adds a single bit (0 &1) to previous query and stores it in a queue (Q). Reader then pops out a query and broadcasts the query. If the reader read any tag (possible only if one tag responds) or doesn t get any response then the query is placed in read queries queue (RQ). This cycle is continued until the Q is emptied and all the queries are placed in RQ; resulting the removal of tag-starvation problem.

Tags on the other hand; if they receive a gap-pulse; each tag checks for a flag if it responded back. If flag is set to true then each tag seeks a time interval and keeps aside. When a tag receives the query from the reader; it checks if query is equal to its ID. If the query is equal and the timer value is zero, then the tag responds to the query. This cycle is repeated whenever it receives a gap-pulse; resulting the reduction of collisions in the process.

Figure 1(a) gives the algorithm for tag operation and figure 1(b) gives algorithm for reader operation. The algorithm for reader includes inserting the queries and deleting them from the queue as queue form the basis for the whole algorithm. The algorithm results in a slower process but provides an efficient solution for the problems being faced in RFID system.

Tag Operation:

```
/* q=q1q2… holds the query received from reader
   n = randomly picked time
   g= gap pulse
   F = flag value
   ti= binary value of ID */
Receive the command starting a frame from reader.
Receive message m from reader.
while m!=frame terminating command do
if m=g and F=1 then
   n=random(time in ms)
else
   if n= system time then
      q=m
      flag=1
      for all i = 0, i< length of q    do
         if qi = ti then
            flag=0
            break
         end if
      end for
      if flag=1 then
          Transmit ID
          F=1
      end if
   end if
end if
end while
```

Reader Operation:

```
/* Q = queue to hold queries after adding 1-bit
   RQ = read queries queue    */
Q = RQ
RQ = NULL
if Q = NULL then
   Push(Q,0)
   Push(Q,1)
 end if
Transmit the frame starting command
while Q!= NULL do
```

```
    q= Pop(Q)
    Transmit q
    Receive tag response and identify if collision occurs
     if tag collision then
          Transmit „gap pulse
          Push (Q,q0)
          Push (Q,q1)
     else if only one tag responds OR no tag responds then
          //funcionalities
          Push (RQ,q)
     end if
end while
Transmit terminating frame
```

V. CONCLUSIONS

The algorithm developed using the ALOHA and QT here, eliminates the tag-starvation problem and also reduces the collisions that are taking place in tag-reader communication. ALOHA is used to reduce the collisions occurring at reader in order to fasten the process of tag identification and also preserves the identification accuracy.The performance of QT protocol is increased indicating the efficiency and effectiveness of the proposed algorithm.

REFERENCES

[1] Chuen-Ching Wang, Te-Yuan Wang, Tien-Hao Lien, Rong-Ming Su, "*A Simple and Efficient Anti-Collision Algorithm for RFID System,"* Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010.

[2] Xinqing Yan and Guiliang Zhu, "*An Enhanced Query Tree Protocol for RFID Tag Collision Resolution with Progressive Population Estimation,"* IEEE 2009, pages: 935 – 940.

[3] Jihoon Myung, Wonjun Lee, Jaideep Srivastava, Timothy K. Shih, "*Tag- Splitting: Adaptive Collision Arbitration Protocols for RFID Tag Identification,"* IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 6, June 2007.

[4] Yan Xin-qing, Yin Zhou-ping and Xiong You-lun, "*QTS ALOHA: A Hybrid Collision Resolution Protocol for Dense RFID Networks*," IEEE International Conference on e-Business Engineering, 2008.