Relay MAC: A Collision Free And Power Efficient Reading Protocol For Active RFID Tags

Gautam Bhanage and Yanyong Zhang

Abstract— A promising application for RFID tags is to trace valuable assets in an inventory. In such systems, the key challenge is to achieve reliable and energy-efficient tag reads. This paper proposes a novel tag reading protocol, *Relay-MAC*, which aims at reducing the information sent over the network and the energy spent in collision detection and handling by introducing deliberate sequencing at runtime. This paper provides an in-depth study of the design issues one may face in implementing such a protocol on RFID tags, and validates its feasibility using simulation studies. These studies clearly demonstrate that Relay-MAC can yield much better throughput and energy conservation when compared to a conventional select-and-read protocol.

I. INTRODUCTION

In this paper, we present *Relay-MAC*, a novel medium access control protocol for active tags to report to the base station. Relay-MAC does not involve the reader in every tag's transaction, instead it uses one tag's transmission to trigger the next tag's transmission. The resulting scenario is analogous to a "relay" in real life. Relay-MAC can efficiently sequentialize the transmissions from all the tags, thus minimizing collisions and maximizing the tag's sleeping periods. Packet transmission time is reduced by employing explicit addressing. This addressing scheme makes use of the less dynamic nature of the application to reduce the information sent over the network. The protocol tries to maintain sanity in the face of realistic network situations such as packet loss and corruption by keeping packet transmissions to a minimum. We have studied the effectiveness of Relay-MAC using a detailed event-driven simulator, and compared its performance with a normal select-and-read protocol [1]. The simulation results have shown that Relay-MAC leads to better reading throughput and longer tag lifetimes.

In the rest of this paper, we provide an overview of the targeted system in Section II, and an overview of the related work in Section III. The detailed design of Relay-MAC is presented in Section IV, and the results are reported in Section V. Finally, the concluding remarks and future directions are discussed in Section VI.

II. SYSTEM OVERVIEW

System Assumptions: We propose our system design based on the operating conditions listed below:

G.D. *Bhanage* and Y.Y. *Zhang* are with the department of Electrical & Computer Engineering at Rutgers University and are affiliated with the Wireless Information Network Labrotory [WINLAB]. Email: [gautamb/yyzhang]@winlab.rutgers.edu

- *Active Tags:* In this study, we focus on active tags (beacons), which have their own internal power source for all the circuitry and the tag radio, and can broadcast their own signal. They have longer ranges and larger memories than passive tags, as well as the ability to store additional information sent by the transceiver.
- *Receiving Capability:* In this paper, we assume the tags are able to pickup signal from the reader as well as other tags. While receiving signals from the reader is easy, the need for tags to hear each other requires the tags not be spread far apart. Otherwise, they should be organized in groups and a relay protocol should be implemented within the groups. We also assume that this network does not suffer from the problem of hidden and exposed nodes.
- *Frequency Range For Applications:* We also assume that a tag's transmitter and receiver operate at the same frequency as the reader and other tags.

III. EXISTING TAG READING PROTOCOLS AND THEIR LIMITATIONS

Standard Anti-Collision protocols can be classified into two broad classes: Contention based protocols and Select and Read protocols.

Contention Based Techniques: Contention based protocols consist of probabilistic protocols like ALOHA [3] and CSMA [4]. In ALOHA, a node simply transmits a packet when it is generated (pure ALOHA) or at the next available slot (slotted ALOHA [5]). Packets that collide are discarded and will be retransmitted later. Frame slotted aloha [6] reduces collision by grouping medium access into frames and having multiple slots within each frame. In this way, tags transmit at most once in a randomly selected slot. Dynamic Frame slotted aloha [7] estimates the number of tags around the reader, and dynamically allocates the frame size for the number of tags. The main problem with ALOHA-based schemes is high collisions or low slot efficiency. CSMA-like techniques can prevent collisions, but they spend a significant amount of energy in listening to the channel in a short range radio [8], [9].

Select-Read And Deterministic Techniques: Tag starvation is another problem associated with contention based protocols where some tags may repeatedly fail to be identified over a certain number of read cycles. This problem can be alleviated by the select and read protocols such as the binary tree protocol [10] and the query tree protocol [11] by explicitly reading all the tags every cycle. One of the major drawbacks of such

techniques,however, is the long identification delays associated with the singulation, partially due to the low bit efficiency in addressing tags. Deterministic TDMA systems could provide a feasible solution, but they are often static and would face synchrony problems with the cheap oscillators on the tags for time multiplexing.

IV. RELAY STYLE MEDIUM ACCESS CONTROL (RELAY-MAC)

In this section, we present the detailed design and working of the Relay-MAC protocol.

A. Basic Protocol

We first present the basic protocol for Relay-MAC assuming a static set of tags and fault-free operating conditions. Later, we discuss how to extend the basic protocol to address several realistic issues such as dynamically joining and leaving tags and faulty network conditions.

Information Maintained by the Reader and the Tags: The reader manages most of the data structures and maintains the global information for all the tags in its range. The main data structure is a list of the tags, referred to as *TagList*, which includes each tag's ID, and the corresponding slot number in which the tag reports to the reader.

In the system initialization phase, the reader sends out a pulse to all the tags, asking them to report their IDs. Each tag can respond to the reader after a random amount of time to avoid collisions. After a sufficiently large time frame, the reader will have all the tag IDs. These Tag ID's are populated in the *TagList*. After obtaining all the IDs, the reader can then allocate sequential slot numbers to each tag in the *TagList*. Later, the reader announces each tag's *TagSlotNumber*, or *TSN* in short, which will be stored in the tag's memory. The tags are programmed in such a way that they use their physical address only when the TSN field in their memory is not defined.

Packet Format: Figure 1(a) illustrates the generic frame format that is used in Relay-MAC. It has seven fields which are explained as follows: The Sync field is a standard bit pattern used to synchronize the radios of the transmitter and the receiver, the u_r and c_r fields represent unconditional and conditional retransmission control bits which are required for handling packet losses. These bits are discussed elaborately in the subsection on handling lost packets. The *TSN* field carries the sequence number of the tag to be identified. The reader circulates the size of *TagList* in the N field to help them decide their sleep duration. The CRC field is a conventional error checking code. Finally, the packet can have a Payload field which would find applications in a sensor network scenario.

Using the generic frame format, Relay-MAC derives two types of packets. The first packet type is called *ReqReport*, and such packets are sent by the reader to request a report from a specific tag. The format of a *ReqReport* packet is illustrated in Figure 1(b). Figure 1(c) shows the second packet type which is called a *Report* packet, and such packets are usually used by the tags to report to the reader and trigger other tags.
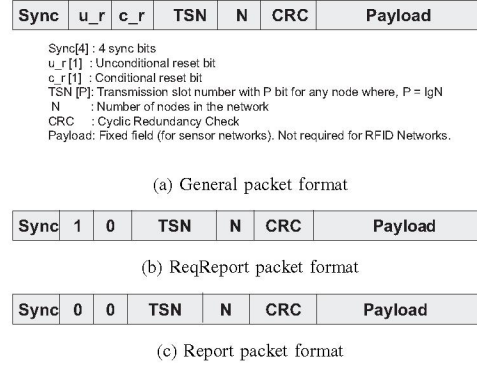

Fig. 1. The generic frame format (a) and two typical packet formats (b) and (c) that are derived from (a) in Relay-MAC.

Protocol Overview: The basic protocol under ideal network conditions has the following stages:

1) *Reader Requests Report:* At the end of the initialization phase, the reader obtains the *TagList*, and it sends out a *ReqReport* packet with the *TSN* of the first tag on the TagList. The u_r bit is set and c_r bit is reset in this packet. This indicates that it is the beginning of the read cycle and the tag whose *TSN* matches with the *TSN* advertised in the packet should respond.

2) *Tags Relay Reports:* After the ReqReport packet reaches the specified tag, within time δ, the tag will assemble a Report packet whose *TSN* field will be set to its *TSN+1* and broadcast it. After sending the packet the tag puts itself to sleep for a duration decided by N in the received packet. The Report packet will be received by the reader and all the other tags in the network. The next tag (in sequence) will detect this packet, identify its *TSN* with that advertised in the packet and determine that it is its turn to transmit. The tag will then assemble a Report packet by incrementing its own *TSN* field, and send it out. If there is no exception, this process should repeat itself, and the reader will receive reports from all the tags every $n_{tag}\delta$ time. The value of $n_{tag}\delta$ is the duration of a read cycle.

Ideally, after a tag transmits its Report packet, it can put itself to sleep for $(n_{tag} - 1)\delta$ time before a retransmission is attempted on that node. In this way, a significant amount of energy can be conserved. However, this is not very realistic because it completely ignores the clock drifts among all the tags. In order to accommodate clock drifts, we let each tag sleep for $a(n_{tag} - 1)\delta$, where $a(0 < a < 1)$ is a constant whose value can be adjusted to reflect different degree of clock drifts. In general, we would like choose a's value close to 1 so that a tag can sleep longer and conserve more energy. On the other hand, a slightly smaller a is preferred to accommodate the crystal drifts. Also, we can let the reader synchronize the clocks on each tag periodically (the interval for clock synchronization should be much larger than a read cycle duration) so that we can adopt a larger a.

In order to determine the duration of a read cycle, let β denote the transmission delay between the two farthest nodes

in the network, and γ the time to process a packet. For reading the first node in the network, the reader has to initiate the read and it is similar to a select and read operation, hence the time taken to read the first tag is $\delta = \beta + \gamma + \beta$. For the remaining N-1 nodes, the response signal of the previous tag will trigger the next tag in sequence, and hence, the read time for each is $\tau = \gamma + \beta$. Hence, the duration of a read cycle D is calculated as

$$D = \delta + (n_{tag} - 1)\tau = n_{tag}\gamma + (n_{tag} + 1)\beta \qquad (1)$$

Thus every node n_k will read the current packet in the network and will send a response packet after $[t+n_k\delta]$, where t is the time when the cycle starts.

B. Extending Relay-MAC to a Dynamic RFID System

In the above discussion of the basic protocol, we assume that the tags in the system are static, and therefore, the number of tags, n_{tag}, is fixed. In this section, we discuss how to extend Relay-MAC to a more dynamic RFID system where more tags may join the reader. Instead of letting every tag in the system re-register with the reader every time when new tags join the system, we propose a *Membership Joining Protocol* involving only the new tags announcing themselves to the reader.

For a realistic asset monitoring application, we take the viewpoint that new tags join a reader every time when more items are added to the inventory. In such cases, the reader can check for new tags every p read cycles. In the *Membership Joining Protocol*, the reader will start the Initialization phase every p read cycles wherein the reader will broadcast a pulse to all the tags. Only those tags without a valid *TSN* field (i.e. the newly added tags) will respond by reporting their IDs after a random delay. The reader will then extend the *TagList* by appending the new IDs to the end of the current *TagList* and allocate a *TSN* to each of the new tags. The process for detecting new tags can be very adaptive. If the number of new tags detected every p cycles remains the same, then the p value remains the same. However if you start detecting more number of tags per p cycles, then this indicates an increase in the arrival rate of new tags in the system. If the arrival rate is detected to be going above a definite threshold then the value of p can be reduced to check for new tags more frequently. If the arrival rate decreases, similar action can be taken to increase p and improve reading throughput. Our protocol works best with a larger p value.

In addition to dealing with new tags, the reader should also handle the loss of tags. When a tag leaves the environment or fails due to battery depletion, the reader will detect repeated *relay-failures* (explained in the next subsection) from the tag in the corresponding slot. Once the kth tag's absence is established, the reader will send out a *ReqReport* packet with the *TSN* of n_{k+1} whenever a packet with the *TSN* of n_k is received. This will smoothen out the reading cycle and does not create a miss for subsequent reads.

C. Extending Relay-MAC to Handle Lost Packets

This section discusses how the basic Relay-MAC protocol may be extended to deal with realistic network conditions such
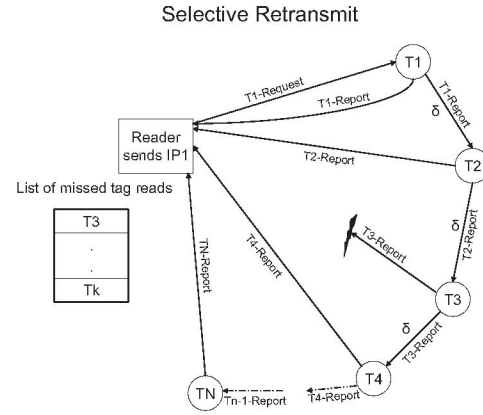

Fig. 2.   Reader misses a packet in one slot.

as packet loss or corruption. In our discussion, we assume that such adverse network conditions are transient, and dealing with permanent problematic network conditions is not the scope of this study. Packet loss is detected by the reader, and there are two generic situations in which this could occur:

1) The reader misses a packet in one slot but receives packets in the subsequent slots.
2) The reader misses packets in two consecutive slots.

In Case 1, the reader misses a *Report* packet in the kth slot, but it receives packets in the $(k+1)^{th}$ slot and the following ones. This is an indication that tag k's packet has reached tag $k+1$, but not the reader. This type of failure is referred to as a *reader link failure* and is shown in the Figure 2. The reader handles this situation by simply saving the missed TSN (k, in this case) in the *MissedList*. The *MissedList* data structure is identical to the *TagList* data structure and is needed to handle lost packets. All the elements in the *MissedList* will be dealt at the end of the current reading cycle by running the *selective-retransmission* protocol. The details in the selective-retransmission protocol will be discussed later in this section.

In Case 2, suppose that the reader misses a *Report* packet in the slots k and k+1. This indicates that some $(k-1)$ tag's Report packet failed to reach its subsequent tag (k) and the entire relay sequence failed. This type of failure is referred to as a *relay failure* and is depicted in Figure 3. In this case the reader records k in the *MissedList* and sends a *ReqReport* packet with $TSN = (k+1)$ in the next slot. Hence the tag with *TSN*, $k+1$ will respond to the reader request and the cycle will start from the $(k+1)^{th}$ tag to the last causing a complete re-transmit.

Note that if both the cases mentioned above happen at the same time due to the non-localized nature of the disturbances, the reader would still detect it as a *relay failure*. There could be a special case where there is an extremely localized disturbance around the reader causing consecutive *reader link failures* which are interpreted by the reader as *relay failures*. However, it is highly unlikely that the disturbance affecting the reader will not affect the tags because the reader is equipped with a more powerful receiver than the tags.
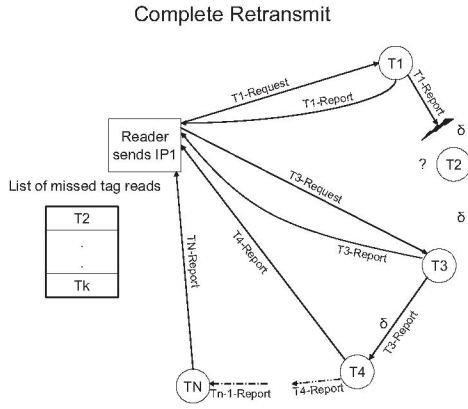
Fig. 3. Reader misses a packet in 2 consecutive slots.

Fig. 4. Selective retransmission packet format.

After building the *MissedList*, the reader performs a selective retransmission on those tags whose slots are on the list. In order to achieve this purpose, we have to make several modifications to the baseline Relay-MAC protocol.

First, we need to employ an *early wakeup* mechanism. Between the end of one read cycle and the beginning of the next, the reader will perform the selective retransmission operation. To accommodate this, each tag should wake up at the end of the read cycle instead of sleeping till its transmission time in the next round (turnaround time). Hence each tag with a *TSN* of k will sleep for a time of $a(n_{tag} - k)\delta$ time instead of the ideal $a(n_{tag} - 1)\delta$ time. After the re-transmission phase the tags could again go to sleep until their turn in the next read cycle.

Second, we need to employ the *DataSent* Bit. The *DataSent* bit is like a two state finite state machine present on each tag which indicates whether a tag has transmitted for the current read cycle or not. Whenever the tags receive a *ReqReport* packet they know that it is the beginning of the read cycle and all the tags reset their *DataSent* bit. Then the tag which had its *TSN* in the packet responds and sets *DataSent*. The next tag in sequence will also notice that its *DataSent* bit is reset and hence it can transmit without checking the u_r and c_r bits in the packet.

Third, we need to introduce the *Selective Retransmission Protocol*. At the end of the read cycle all the tags have woken up and the reader is ready to handle the retransmissions. The reader sends a retransmit packet to each *TSN* in the *MissedList*. The format of the Selective Retransmit packet is shown in Figure 4. Upon receipt of the Retransmit packet, each tag checks for the *DataSent* bit. If the tag had a reader link failure, it has its *DataSent* bit set. But if it has a relay failure, the *DataSent* bit is reset as the tag never receives its trigger packet. If the *DataSent* bit is reset, the tag responds unconditionally. However, if the *DataSent* bit is set, the tag checks the retransmission bits on the Request packet to determine whether

```
Packet P = getpacket()
If P == InitPacket()
    If(TSN Not defined)
        SendTagId()
//wait for reader to allocate TSN.
Packet P = getpacket() myTSN = getTSN(P)
while(1) {
  Packet P = getpacket()
  add = getTSN(P)
  if(add == myTSN)
  {
  t = getSleepTime(P)
  controlData = getControl(P)
  P = responsePacket(nextAdd,controlData,sleep)
  Broadcast(P)
  sleep(t)
  }
}
```

Fig. 5. Algorithm Running On the Tag/Node

it is eligible to transmit in the next slot. Since the retransmit packet has this bit set, the tag tries to verify its *TSN* with the packet. If there is a match the tag replies with a response packet (which has the retransmit control bits reset). Since the Tags that transmitted successfully in the read cycles have their *DataSent* bit set and they receive a packet with both the retransmit control bits reset, they ignore the incoming packet. Hence this packet will fail to trigger a 'relay' response from the remaining nodes and the reader can read individual tags at the end of the read cycle. The tags which re-transmit do not go off to sleep as the packet from the reader indicates that this is a selective re-transmit and the tags might have to retransmit again if the read fails.

The case of node failure will be handled in the same way as the case when a node moves out of the reading area. The reader will detect repeated failure for that tag read and will eventually remove it from the *TagList*.

V. PROTOCOL PERFORMANCE AND DISCUSSION

In this study, we have implemented a discrete event-driven simulator to study the performance of Relay-MAC and compare it with the select and read protocol. We report the performance results in this section.

A. Relay-MAC vs. Select-and-Read Protocol

The select-and-read protocol is a typical example of Reader Talks First protocol in which the reader announces the desired tag ID for every read operation. In the discussion below, let us assume that β and γ have the same values as used in Equation 1. Therefore, each operation in the Select and read protocol can be characterized by a *Reader-to-tag transmission delay* (β), *Tag processing delay* (γ) and consequently the *Tag-to-reader transmission delay* (β).

Hence the duration of a read cycle for the select-and-read protocol is

$$D' = n_{tag}(2\beta + \gamma). \tag{2}$$

From Equation 1, the duration of a Relay-MAC read cycle is
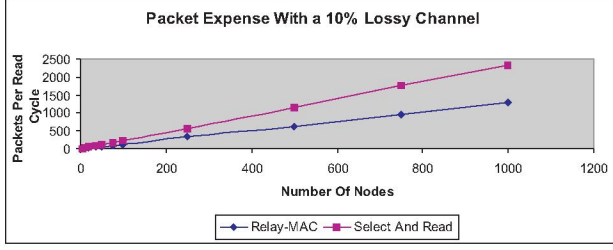
$$D = n_{tag}(\beta + \gamma) + \beta. \tag{3}$$

Fig. 6. Performance comparison of the read cycle duration of a Normal select and read protocol and of the proposed Relay-MAC protocol for different tag count, with packet loss probability being 10%.
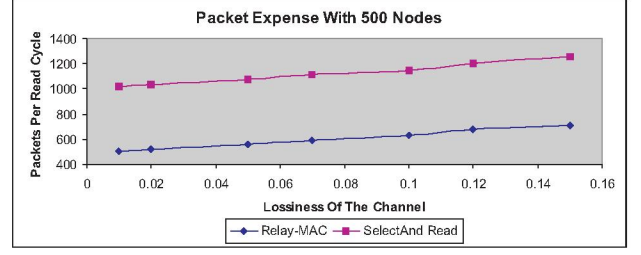

Fig. 7. Performance comparison of the read cycle duration of a Normal select and read protocol and of the proposed Relay-MAC protocol for different packet loss probabilities, with number of tags being 500.

As a result, the ratio of these two read cycle lengths is

$$\rho = D'/D \tag{4}$$
$$= n_{tag}(2\beta + \gamma)/n_{tag}(\beta + \gamma) + \beta \tag{5}$$

Given the fact that $\beta \gg \gamma$, the ratio can further converge to

$$\rho = 2n_{tag}/(n_{tag} + 1) \tag{6}$$

The ratio in Equation 6 gives us a rough comparison of the reading throughput of the two protocols, with the assumption that the β values in these two protocols are the same. In reality, the β value in the select-and-read protocol is actually larger than the value in Relay-MAC. This is due to the fact that the select-and-read protocol employs the 128bit/96bit tag address in their packets, while Relay-MAC employs the tag slot number (*TSN*) in their packets. For example, the number of bits in Relay-MAC slot number could be $\log_2 n_{tag}$. Hence, the ρ value will be even higher than that indicated in Equation 6.

B. Simulation Results

This study focuses on the reading throughput of different tag reading protocols as well as their energy costs. Specifically, the reading throughput of a protocol is inversely proportional to its read cycle duration. Further, the duration of a read cycle can be measured by the number of packets transmitted within a cycle because transmission delay is the dominant factor for the cycle duration. On the other hand, the energy consumption is measured by the resulting duty cycle of the protocol.

Since the focus of this study is not on the physical channel characteristics, we have employed a simple constant delay to model the transmission delay between any two nodes. In addition, we also assume that each tag incurs a constant processing delay. The lossy channel is modeled as a gaussian process by comparing a randomly generated fraction with the *packet loss probability p* to determine dropped packets. Using this simulator, we have varied the number tags from 3 to 1000 and p from 1% to 15%.

Reading Throughput: Results of the simulations to measure the length of the read cycle with varying number of tags and a fixed link failure probability are shown in Figure 6. The packets sent over the network due to re-transmission are also accounted for in this analysis. Figure 6 shows that Relay-MAC
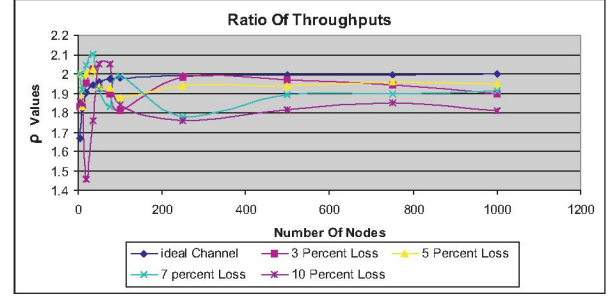
Fig. 8. Reading throughput ratio (ρ) of Relay-MAC with respect to the throughout of a normal select-and-read protocol with different packet loss probabilities and tag counts.

clearly leads to much shorter read cycles and the performance difference between these two protocols increases with the number of tags. The explicit sequencing and packet relay in the proposed architecture drastically reduces the number of packets sent over the network and leads to a better performance despite re-transmissions. These results suggest that as the number of tags attached to a reader increases, the benefit of Relay-MAC will be even more pronounced, which shows that this protocol exhibits good scalability.

We have also studied the impact of the link loss probability on the packet expense of the two tag reading protocols, and the results are presented in Figure 7. As indicated, performance of both the protocols decays as the noise in the channel increases. However the performance of the Select and Read protocol decays at a rate which is almost 12% faster than the performance decay rate of the Relay-MAC protocol when the noise in the channel increases from 1% to 15%. This can be attributed to the fact that, the Relay MAC has fewer number of packet links in the network which leaves a lesser chance for loss and corruption than the Select and Read Protocol. These results clearly show that Relay-MAC is more robust to lost packets and transmission failures.

Figure 8 plots the ratio between the number of packets used in the Select and read protocol to the number of packets used in the Relay-MAC architecture for varying loss probabilities. It can be observed that in the ideal case (with zero packet loss), this ratio is close to 2. Hence the average number of packets used in the Select and read protocol are twice as much as those
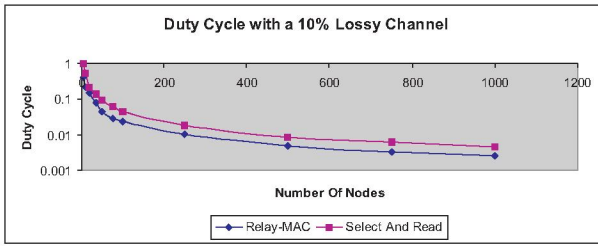
101

Fig. 9. Performance comparison of average tag duty cycle of a Normal select and read protocol and of the proposed Relay-MAC protocol for different tag count, with packet loss probability of 10%
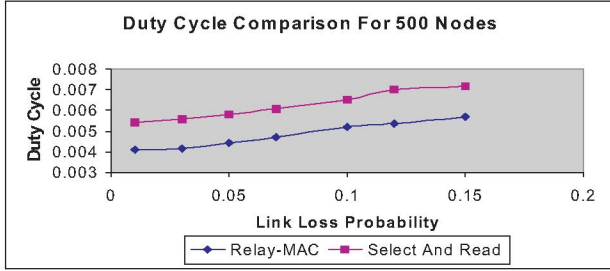

Fig. 10. Performance comparison of the average tag duty cycle of a Normal select and read protocol and of the proposed Relay-MAC protocol for different packet loss probabilities, with 500 tags.

used in the Relay architecture. From the Figure 8, it can also be observed that with an ideal channel the initial performance difference ratio (ρ) is close to 1.6, which can be attributed to the number of packet links in the network which are far too less for the Relay MAC to show a significant advantage (This can be verified by Equation 6). It can also be observed that as the loss in the link increases the average value of the ratio decreases. This is due to the fact that despite the increase in the packets sent over the network with the increasing link loss probability, the rate of decay of performance (increase of packets) for the Select and read network is not exactly twice as much as the Relay architecture. Despite these observations, the performance of the Relay-MAC architecture is still superior to the conventional select and read protocol due to its inherent nature to use approximately half the packets than select and read.

Energy Efficiency: The lifetime of an RFID system is largely dependent on the energy efficiency of the employed tag reading protocol. The energy efficiency of a protocol, is in turn measured by the average duty cycle for a tag, i.e. the number of slots a tag has to be awake per read cycle. Though most of the existing select-and-read protocols require every tag to wake up for every read operation, for the sake of fairness, we have implemented the select-and-read protocol in such a way that a tag goes off to sleep after two time slots of its transmission. It has to stay awake for this period to listen for re-transmission requests. In this way, the duty cycle of a tag in the select-and-read protocol can be significantly reduced.

Figure 9 plots the duty cycles of the two protocols with a varying tag count and a 10% lossy channel. It can be observed that, as the number of nodes in the network increases, the duty cycle decreases. This can be attributed to the fact that as the number of nodes increase, the turn-around time for each node increases, allowing it to sleep for longer periods of time. Despite the developments proposed for the select and read protocol, it still has a higher duty cycle due to the fact that the tags in sequence have to remain awake for two extra cycles per transmission to check whether to retransmit.

Finally, Figure 10 shows the duty cycle as a function of the link loss probability. It can be seen from this figure that as the link loss probability increases, more number of packets are going to fail transmission. Hence in order to handle these re-transmissions, the tags would need to be awake for a longer period of time at the end of the cycle. This will cause the duty cycle to increase as the loss probability increases. As it can be observed from the Figure 10, the duty cycle for the Select and Read protocol grows almost 8% faster than the RelayMAC with increasing link failures. Note that the slot times themselves in each of the protocols are different. Relay-MAC has smaller slot times than those in Select and Read due to the use of *TSN* instead of physical addresses.

VI. CONCLUSION

In this paper we review the requirements for a protocol meant for asset tracking type applications and show the conditions for operation. We then propose RelayMAC, as a solution which exploits the typical characteristics of this problem and tries to provide an optimal solution. Finally, we compare Relay-MAC with standard Select and Read type protocol using event driven simulation. Based on the simulation results, we conclude that the Relay-MAC protocol can provide better performance in a slowly changing network environment.

REFERENCES

[1] M. J. Markowski and A. S. Sethi, *Wireless MAC Protocols for Real-Time Battlefield Communications*. University of Delaware, DCIS, Newark, Delaware, USA. http://citeseer.nj.nec.com/4707.html, 1997.
[2] P Sorrells, *Passive RFID Basics*, MT Inc - Microchip Technology Inc, 1998.
[3] Norman Abramson, *Development of the ALOHANET.*, IEEE Transactions on Information Theory, vol. 31, no. 2, pp. 119 − 123, Mar. 1985.
[4] Leonard Kleinrock and Fouad Tobagi, *Packet switching in radio channels: Part I*, IEEE Transactions on Communications, vol.23, no. 12, pp. 1400 − 1416, Dec. 1975.
[5] Gildas Avoine and Philippe Oechslin, *RFID Tracability a Multilayer problem*, EPFL Lausanne, Switzerland, pp. 125 − 140, 2005.
[6] D. Hush and C. Wood, *Analysis of Tree Algorithms for RFID Arbitration*, IEEE International Symposium on Information Theory, 1998.
[7] J. Cha and J. Kim, *Novel Anti-collision algorithms for fast object identification in RFID systems*. 11th International Conference on Parallel and Distributed Systems, pp. 63 − 67, 2005.
[8] Wei Ye and John Heidemann, *Wireless sensor networks*, pages: 73 − 91, ISBN:1-4020-7883-8, 2004.
[9] M. Stemm and R. Katz, *Measuring and reducing energy consumption of network interfaces in hand-held devices*, IEICE, vol. E80-B, no. 8, pp. 1125 − 1131, 1997.
[10] Jihoon Myung and Wonjun Lee, *An Adaptive Memoryless Tag Anti-Collision Protocol for RFID Networks*, INFOCOM, 2005.
[11] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, *"Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID systems*, ISLPED, August 2004.
[12] Christian Floerkemeier and Matthias Lampe, *Issues with RFID usage in ubiquitous computing applications* Second International Conference on pervasive computing, Austria, 2004.