Collision Avoidance in a Dense RFID Network

Shweta Jain
Computer Science Department
Stony Brook University
Stony Brook, NY
shweta@cs.sunysb.edu

Samir R. Das
Computer Science Department
Stony Brook University
Stony Brook, NY
samir@cs.sunysb.edu

ABSTRACT

In this work, we develop a CSMA-based MAC protocol to avoid reader-reader and reader-tag collisions in a dense RFID network. The network is implemented using mote-based RFID readers. To implement the MAC protocol, we develop an appropriate carrier sensing circuit using an RFID tag as an antenna and the mote as an apparatus to sample received signal strength. We have augmented a commercially available OEM RFID module with such carrier sensing capability and interfaced it with motes. Performance evaluation shows much superior performance relative to a naive and a randomized protocol in dense deployment environments both in regards to accuracy and time per tag read.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communication, Wireless communication*

General Terms

Design, Experimentation, Performance

Keywords

RFID, CSMA MAC Protocol

1. INTRODUCTION

RFID (radio frequency identification) [1] is an automatic identification system that consists of two components – readers and tags. A tag has an identification (ID) stored in its memory that is represented by a bit string. A reader is able to read the IDs of tags in the neighborhood by running a simple link-layer protocol over the wireless channel. In a typical RFID application, tags are attached or embedded into objects in need of identification or tracking. In the most common application of RFID (e.g., supply-chain management), RFID tags simply serve the purpose of UPC bar codes. By reading all the tag IDs in the neighborhood and then consulting a backend database that provides a mapping between IDs and objects, the reader learns about the existence of corresponding objects in the neighborhood. This way RFID readers also act as identification and/or proximity sensors.

RFID tags can be either *active* or *passive* depending on whether they are powered by battery. We are interested in passive tags in this work. Passive tags are prevalent in supply chain management as they do not need a battery to operate. This makes their lifetime unlimited and cost negligible (only few US cents per tag). The power needed for passive tags to transmit their IDs to the interrogating reader is supplied by inductive coupling between the reader and tag antennas. The reader "energizes" the tags in the vicinity with RF power continuously for the entire read operation. In the most prevalent form of the technology, part of this power is used to transmit a response back to the reader (using a process called *backscattering*) after appropriate modulation and coding via the tag's electronics.

While RFIDs have mostly been used in supply chain management so far, our interest in this work is studying their performance in a very dense deployment scenario as will be common in "smart environment" applications. In such applications, we envision that there will be a lot of tiny readers deployed in a dense fashion – much like a sensor network – observing the tagged environment around them by reading tags continuously or periodically. There will also be a lot of tags around in such environments. This will certainly be the case in smart home or office scenarios as RFID tags will soon replace the UPC bar codes for any item we buy in stores.

However, several collision problems might occur when multiple readers are used within close proximity of each other. Thus, the concurrent read operations must be coordinated appropriately. We will elaborate on these problems in the following section. Current generation RFID systems do not address the multi-reader coordination problems effectively because of their emphasis on supply chain where multiple readers are rarely used in the same physical space.

In this work, we design and evaluate a simple carrier sense-based MAC protocol to avoid collisions in multi-reader scenarios. We build it specifically for a tiny Berkeley mote-based platform [2] for deployments in smart environment applications. The goal of this paper is to describe the design choices we made, the protocol operation and preliminary performance results. The key feature of this design is the use of an RFID tag antenna as an apparatus to measure receive signal strength and the mote platform to sam-

WiNTECH'06, September 29, 2006, Los Angeles, California, USA.

ple it. While many other sophisticated solutions (e.g., use of TDMA-based approaches or multiple frequencies) are possible, the approach we present is simple, requires a bare minimum of electronics to build and performs effectively.

The rest of the paper is organized as follows. We first present a background on the collision problems in RFID systems in Section 2. We then present our system design in Section 3, followed by the description of the MAC protocols in Section 4, their performance evaluation in Section 5 and concluding remarks in Section 6.

2. COLLISIONS IN RFID SYSTEMS

Simultaneous transmissions in RFID systems lead to collisions as the readers and tags typically operate on the same channel. Three types of collisions are possible.

2.1 Tag-Tag Collision

Tag-tag collision occurs when multiple tags respond to the same reader simultaneously. Due to multiple signals arriving at the same time, the reader may not be able to detect any tag. This problem prevents the reader from detecting all tags in its interrogation zone. A popular solution to this problem is the tree walking algorithm (TWA) [3], which is generally used in UHF readers. In this protocol, the reader splits the entire ID space into two subsets and tries to identify the tags belonging to one of the subsets, recursing along the way until a subset has exactly one tag or no tags at all.

Due to larger turn around times at lower frequencies, TWA is not deemed suitable for HF readers. Instead, the HF readers use a slotted termination adaptive collection (STAC) protocol [4] somewhat similar to the framed Aloha protocol. In STAC, tags respond at randomly selected slots whose beginning and end are controlled by the reader. The reader sends a "begin round" command with the number of slots in the round. Tags that are energized by the reader select a random slot number as the proposed reply slot and set their states to "slotted read" and counters to zero. This counter advances each time the reader sends an "end slot" command. A tag sends its response to the reader when its counter reaches the proposed reply slot. If the reader does not hear any tag in a slot, it sends a "close slot" command, which causes all tags to increment their counters. If the reader receives a response correctly, it closes the slot by issuing a "fix slot" command which makes all tags to increment their counters and prompts the tag that was correctly heard to go into the "fixed slot" state, after which the tag responds at this same slot in each round. If however, the reader hears a collision, it sends a "close slot" command forcing all tags to increment their counters, while those tags that had responded in this slot, realize that there was a collision since they did not receive the "fix slot" command, and thus they select another slot for transmission. The RFID reader we will uses in our experiments uses this STAC protocol in the MAC layer for resolving tag-tag collisions.

2.2 Reader-Tag Collision

Reader-tag collision occurs when the signal from a neighboring reader interferes with tag responses being received at another reader. This problem has been studied in the EPC-Global Class1 Gen1 and Gen2 standards for UHF readers [5] [6]. In Gen 1 standard, the reader-tag collision problem is mitigated by allowing frequency hopping in the UHF band or by time division multiple access. In Gen 2 the readers and tags operate on different frequencies so that the tag response does not interfere or collide with reader signals. Either solution requires fairly sophisticated technology.

2.3 Reader-Reader Collision

A reader-reader collision occurs when a tag hears multiple readers at the same time. In this situation, the tag might be unable to respond to any reader at all.

Colorwave [7] is one of the first works to address reader-reader collisions. In particular, it considers an "interference graph" over the readers, wherein there is an edge between two readers if they could lead to a reader-reader collision when transmitting simultaneously, and tries to randomly color the readers such that each pair of interfering readers have different colors. If each color represents a time slot, then the above coloring should eliminate reader-reader collisions. If conflicts arise (i.e., two interfering readers pick the same color or time slot), only one of them wins (i.e., sticks to the chosen color), the others pick another color again randomly. In [8], the authors suggest coloring of the interference graph using k colors, where k is the number of available channels. If the graph is not k-colorable using their suggested heuristic, then the authors suggest removal of certain edges and nodes from the interference graph using other heuristics which consider the size of the common interference regions between neighboring readers.

In this work, we suggest the use of classic carrier sensing techniques and develop a simple CSMA-based MAC protocol to resolve both reader-reader and reader-tag collisions. Much of the above solutions require sophisticated use of time synchronization or a large number of frequencies. They are not practical in tiny low-cost HF readers to be used in smart environment applications. Instead we show that simple carrier-sensing circuits can be built cheaply and effectively that can resolve both reader-reader and reader-tag collisions. For tag-tag collisions, built-in link layer protocols such as STAC can be used without change.

3. SYSTEM DESIGN

In this section, we present the hardware design for a RFID reader that uses carrier sensing to avoid collisions. This system consists of an OEM RFID reader module, a host micro-controller and a received signal strength indicator.

3.1 RFID Reader Module

We use the SkyModule^{TM}M1-mini [9] multi-protocol 13.56 MHz OEM RFID reader module for our work. The read range of this reader is up to 7cm with the internal antenna. The actual range is somewhat dependent on the size of the tag antenna and also the tag orientation. It can read upto 20 tags in a second. It is capable of communicating with a host micro-controller over the TTL, SPI and I2C interfaces. The reader module is capable of responding to ASCII and binary commands sent by the host micro-controller. It can select, read and write RFID tags. The host controller can also read and write the reader's memory and system registers to put the reader in low power sleep mode and to wake it up from sleep. The small footprint and low power requirement makes it suitable for being integrated with the processor radio modules used in RFID-based sensor networks.

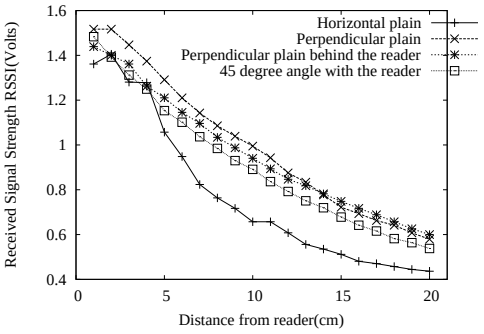


Figure 1: Received signal strength vs. distance between a reader transmitting RFID commands and our RSSI circuit.

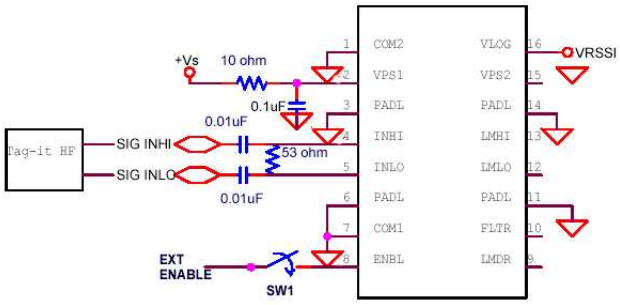
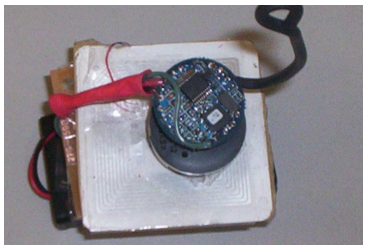
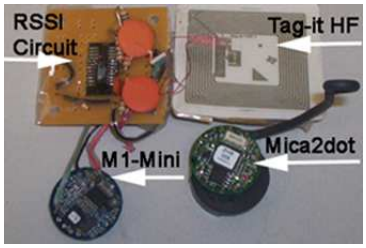


Figure 2: Circuit diagram for the received signal strength indicator (RSSI) circuit.



(a) RFIDMote.



(b) Components of the RFIDMote shown separately.

Figure 3: RFIDMote and its components.

3.2 Host Micro-controller

We have interfaced the Skyetek RFID reader to a mica2dot processor radio module. Mica2dot is based on the well-known Berkeley mote architecture [2] and is manufactured by Crossbow technologies [10]. Equipped with Atmel's Atmega128L 4MHz, 8 bit micro-controller and Chipcon's CC1000 radio, mica2dot can communicate with the RFID reader module via the TTL interface and with the central computer over a 433 or 900MHz wireless link. This setup enables untethered communication between a central controller and the RFID readers. Mica2dot can be programmed with the TinyOS operating system [11, 12].

3.3 Received Signal Strength Indicator

Much of our work has centered around building and experimenting with this module. SkyeModule^{TM}M1-Mini uses a Texas instruments TI-S6700 multi-protocol transceiver. This transceiver does not provide received signal strength of the signal received from tags or neighboring readers. Since we could not obtain the received signal strength directly from the reader, we have built a signal strength indicator circuit that can provide an accurate estimate of the signal strength received from other readers in the neighborhood. This signal strength indicator is later used by the MAC protocol designed to avoid reader-reader and reader-tag collisions.

The Tag-it RFID tag manufactured by Texas Instruments is used to measure the signal level at any point in the reader antenna system. It is often used as charge level indicator to design reader antenna [13] by simply removing the IC from the tag. When the tag is brought in the RF field of a reader's antenna system, a voltage is induced in the parasitic capacitor on the tag. This is a high frequency sine wave whose amplitude varies with the amount of voltage induced in the tag's antenna due to the reader's RF field. In order to measure this signal amplitude accurately, we use an IF limiting amplifier that takes this signal as input and provides a steady voltage as a logarithmic (in db) measure of the input signal amplitude. This voltage can serve as the received signal strength indication (RSSI). We have used the AD8306 chip [14] as the high precision limiting-logarithmic amplifier. The chip provides a perfect linear relationship between the output voltage and the input signal level in db. We connected the output from the charge level indicator (Tag-it HF RFID tag) as a differential input on SIG_{INHI} and SIG_{INHLO} of the circuit shown in Figure 2. The RSSI voltage as measured by this circuit is available at V_{RSSI} and can be sampled by an ADC (analog to digital converter) to "sense" the presence of an active reader in the neighborhood. We use one of the mica2dot's ADCs for this purpose.

To understand the characteristics of our prototype, we measured the variation of the RSSI values obtained from this circuit with distance from an active reader. The results (Figure 1) show that the RSSI progressively diminishes with distance from the reader as expected. We performed this experiment with the RSSI circuit moving away from the reader in the perpendicular plane with respect to the reader antenna. We did this for both sides of the reader. We also moved the RSSI indicator sideways from the reader antenna, i.e., in the same plane as the reader antenna. We measured the RSSI at an angle of 45° with respect to the reader's antenna as well. This set of experiments indicate that the radiation pattern from the reader's antenna is not perfectly omni-directional.

3.4 RFIDmote

The RFID reader module is connected to mica2dot mote that serves as the host micro-controller and communicates with it via the TTL interface. The output of the RSSI circuit described above is connected to ADC2 on the mica2dot and the PW0 port on mica2dot provides the external enable switch to the RSSI circuit. Thus, when the received signal strength is needed, the PW0 port provides the voltage to enable the RSSI circuit and the signal strength is obtained by sampling on ADC2. The RFID reader module, mica2dot mote and RSSI circuit together form the complete system that we have used to evaluate the proposed MAC protocol. We will henceforth refer to this complete system as the **RFIDMote** (Figure 3).

3.5 Power Consumption

Since the target application is an RFID sensor network with battery driven RFIDmotes, power consumption is an important design consideration. The RFIDMote is powered using a 3V power supply consisting of two AA size batteries. We have measured that the the RSSI circuit consumes 14 mA current when it is turned on by applying a voltage on the external enable switch. The RSSI circuit is turned on only when the RFIDMote needs to sense the carrier before instructing the reader to start a new transmission. The RFID Reader module consumes 10 mA current when it is in the idle mode, 60 uA in sleep mode and 60 mA when scanning for tags. Since the RFID reader takes about 100ms to wake up from the sleep mode, we keep the reader in IDLE mode at all times, except if the RFIDMote is itself in sleep mode.

The mica2dot can operate at a low power mode with the radio turned off (8 mA current consumption) or in a sleep mode($\leq$ 1uA current consumption). The radio is turned on only when the RFIDMote needs to communicate tag data. The radio consumes 27 mA in the transmit mode and 10 mA in the receive or idle mode.

Based upon these known or measured values we estimate the current consumption of RFIDMote in various states and tabulate the results in Table 1. A designer can use these values as a guidance for protocol design. Note that channel sensing (i.e., sampling RSSI values) is much less expensive than scanning for tags. Given that the channel sensing is only momentary relative to scanning for tags, channel sensing can provide valuable energy savings as it eliminates wasteful scanning.

4. PROTOCOLS

We implemented three protocols to evaluate tag reading performance in a multi-reader environment. These three protocols – *naive protocol*, *random protocol* and *CSMA protocol* – are discussed in this section. Since we do not have control over the reader firmware, we have implemented these protocols in RFIDMote in software using TinyOS.

4.1 Naive Protocol

In the naive protocol, the RFIDMote transmits a reader-tag inventory request at constant intervals. If two readers are placed in such a way that their interrogation zones overlap, it is possible that some tags would escape detection due to collision (reader-reader collision). Also if two readers are active at the same time and they are close to each other, the signal from one reader would interfere with the tag responses received from the other (reader-tag collision). Since the readers send commands at the same fixed intervals, these collisions may be repeated and it is possible that some tags are never read by any reader. This is a naive reading procedure and is quite prone to reader-tag and reader-reader collisions.

We implement this protocol on the mica2dot using TinyOS. The mica2dot starts a timer using the `call Timer.start (TIMER_ONE_ SHOT, interval)` command and when `event Timer.fired()` is signaled, the mica2dot sends a "read" command to the reader via the TTL interface. The reader now attempts to read the IDs of all tags in its interrogation zone. In this mode, the reader executes the STAC anti-collision protocol, to prevent tag-tag collision discussed earlier. When the reader gets a tag response, it sends the response to the mica2dot via the TTL interface. When all tags have been read, the reader sends a special "read complete" command to indicate that it has completed the execution of the anti-collision protocol and there are no more tags to be read. When the mica2dot receives the "read complete" command, it stores the tag IDs read by the reader. The central computer polls each RFIDMote one at a time to receive the tags read by the readers.

4.2 Random protocol

The naive protocol is prone to reader-reader and reader-tag collisions. A simple method to reduce the chances of collision is the introduction of randomization in the reading schedules. Thus, if the readers choose to backoff for a random interval before sending a read command, the probability of collision may be lower. We introduce a random access protocol in which the mica2dot in RFIDMote, sends a read command to the reader after waiting for a random interval. In TinyOS this random interval is generated by using the `RandomLFSR` component. The size of the window may be varied by masking the 16 bit random number generated via the `RandomLFSR` component. Thus, if the desired window size is 2^7ms, we mask the random number by a bitwise AND with 0x3F. When the mica2dot on the RFIDMote is ready to send a read command to the reader, it goes into a random backoff state by starting a timer for a random duration by executing `call Timer.start (TIMER_ONE_SHOT, (call Random.rand()) & cw)`, where, `cw` is the masking integer to limit the value of the generated random number within the desired window size. When `event Timer.fired()` is signaled, mica2dot sends a read command to the reader, which then immediately starts the RFID transmission. Since the RFIDMotes choose to send commands after random intervals, the commands from two readers would not be concurrent with high probability, given that the window size is sufficiently large. In case there is a collision, it is less likely that the collision will recur for subsequent read commands because the RFIDMote re-selects the interval each time it sends a command.

4.3 CSMA Protocol

Here, when the mica2dot on the RFIDMote is ready to send a read command to the reader, it starts a backoff timer for a random interval, by executing `call Timer.start (TIMER_ONE_SHOT, (call Random.rand()) & cw)` command. Meanwhile, the mica2dot continuously samples the voltage on ADC2 to which the RSSI circuit is connected. If the

Table 1: Power Consumption of RFIDMote at 3V.

RFIDMote State	Mica2dot State	RFID Reader State	RSSI Circuit State	CC1000 Radio State	Current Consumption(mA)
Sleep	SLEEP	SLEEP	OFF	OFF	0.007
Idle	IDLE	SLEEP	OFF	OFF	8
Ready	IDLE	IDLE	OFF	OFF	18
Sensing RFID channel	IDLE	IDLE	ON	OFF	32
Scanning for tags	IDLE	SCANNING	OFF	OFF	68
Transmitting data	IDLE	SLEEP	OFF	Transmitting	35
Receiving data	IDLE	SLEEP	OFF	Receiving	18

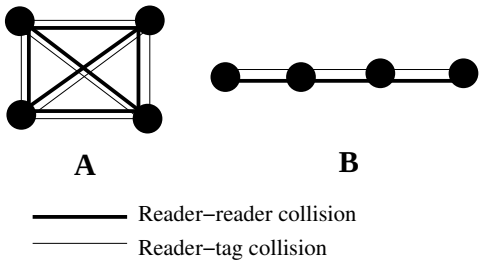


Figure 4: Conflict graphs for (A) square grid and (B) straight line configurations.

voltage read from the ADC is less than a threshold voltage throughout the backoff interval, i.e., until `event Timer.fired()` is signaled, mica2dot sends the "read" command to the reader. In case the mica2dot senses that the medium is busy, i.e., it reads a voltage higher than the threshold voltage on the ADC2 port, it stops the timer by issuing the `call Timer.stop()` command which prevents `event Timer.fired()` from being generated. The mica2dot then continues to sense the medium and when the medium becomes free and stays free for a random duration between 1 and 16 ms, it restarts the timer. This carrier sensing and backoff procedure, enables the RFIDMote to make a more informed decision about scheduling the RFID transmission, that in turn further reduces the chances of reader-reader and reader-tag collisions.

A note is due on the choice of threshold voltage. We have observed that reader-reader collisions occur when the voltage read from the ADC is greater than 1 V which corresponds to a maximum distance of about 10 cm between the readers. The reader-tag collisions occur at a slightly higher voltage, when the two readers are about 5 cm apart. At this distance, the tag may be able to receive signals from both readers. Thus, to solve both reader-reader and reader-tag collisions, we chose the lower of the two, i.e., 1 V as the threshold voltage.

5. PERFORMANCE EVALUATION

We will now discuss the experimental setup and analyze the performance of the RFIDMotes with the protocols discussed in the previous section. We have used accuracy and time taken per read as two performance metrics in our experiments. Let us first define these metrics.

DEFINITION 1 (ACCURACY). *Accuracy is the ratio of the number of unique tags read by all readers to the total number of tags in the interrogation zone of all the readers.*

In order to compute the accuracy of the system we need to determine the number of tags in the interrogation zone of the readers. We activated readers one at a time and allowed them to read all the tags in their respective interrogation zones without any interference from other readers. We recorded the number of unique tags that were read by all readers in each experimental setup. This is the maximum number of tags in the entire interrogation zone. We then use this number for calculating accuracy.

DEFINITION 2 (TIME PER READ). *Time per read is the ratio of the maximum time taken to complete all reads to the number of tags read.*

The maximum time is the time taken by the reader that finishes last and the number of tags read is the total number of unique tags read by all readers. Time is calculated from the point the RFIDMote starts the timer before sending the read command to the RFID reader and until the reader sends the "read complete" response indicating that there are no more tags to read.

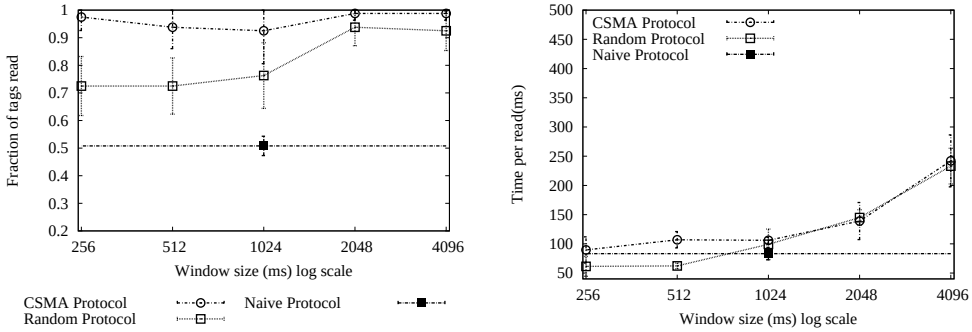
5.1 Experimental Setup

We built and programmed four RFIDMotes and arranged them in different configurations (or topologies) to experiment with the protocols described before. There are 25 tags distributed uniformly in the area . The experiments are controlled by a central computer that broadcasts commands to the RFIDMotes to run specific protocols with specific parameters (e.g., window size) and collects results at the end of the experiments. Each individual experiment is repeated 20 times and average performance metrics are presented. For protocol comparison identical configurations (RFIDMotes and tags) are used.

It is expected that the performance of the protocols will be influenced by the density of the RFIDMotes as this influences how probable the collisions are. Thus, for each configuration we experiment with we show a conflict graph to demonstrate what types of collisions are likely. The conflict graph shows an edge between two nodes (RFIDMotes) that can potentially collide. A thick edge is drawn to denote reader-reader collision and a thin edge is drawn to denote reader-tag collision. The conflict graph is determined via a separate experimental evaluation. More edges in the conflict graph means more gain from the use of carrier sensing.

5.2 Results

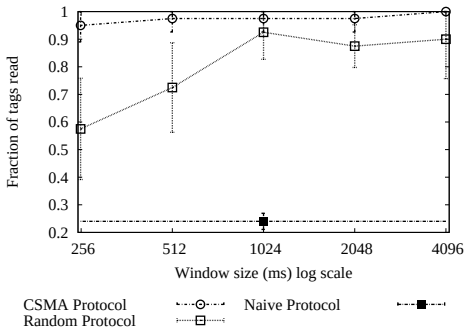
We first show the results of some hand created topologies. We placed four RFIDMotes very close to each other in a square. The conflict graph of this topology is shown in Figure 4A. This is a dense topology in which all readers collide



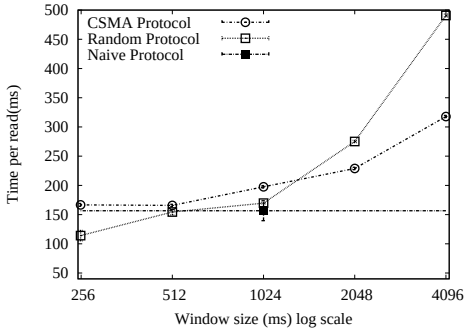
(a) Accuracy.

(b) Time taken per read.

Figure 6: Accuracy and time taken per read vs window size for four readers in a straight line.



(a) Accuracy.



(b) Time per read.

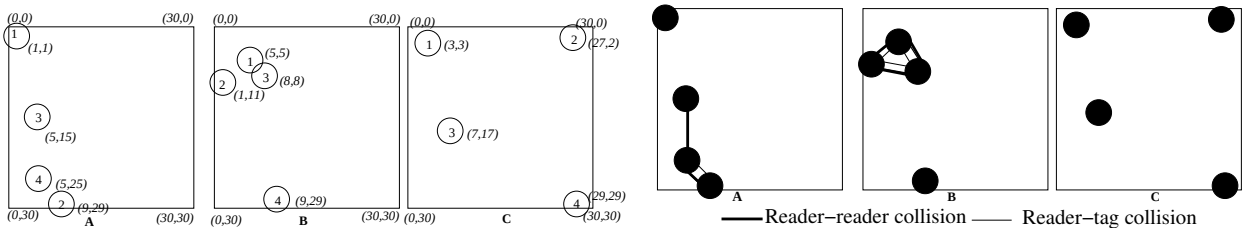
Figure 5: Accuracy and time taken per read vs. window size for four readers in a square grid.

with one another. We measured the accuracy and time per read. The results along with the 95% confidence interval are shown in Figures 5(a) and 5(b) respectively. The horizontal axis shows the varying window size for random and CSMA protocols and the vertical axis shows the accuracy and time per read for each protocol. The naive protocol is shown as a straight line since window size is not a parameter here. The accuracy graph shows that the CSMA protocol achieves much better accuracy than the naive protocol. It is much better than the random protocol when the window size is small. The random protocol improves when the window size is increased, which is obviously due to the increase in the diversity of intervals chosen by each RFIDMote due to larger window size. This improvement comes at the cost of longer time taken to read each tag as seen in Figure 5(b).

We then placed four readers in a straight line. The conflict graph for this setup is shown in Figure 4B. We plot the accuracy and time consumed in reading each tag in Figures 6(a) and 6(b) respectively. This is a less dense topology compared to the grid before and only the adjacent readers can collide. This is the reason why the naive protocol is now able to read more tags than before, but the accuracy still remains poor compared to the random protocol. The CSMA protocol, still performs much better than the rest. Here, we notice that at smaller window sizes, the time taken per tag by CSMA is larger than the random protocol. The reason for this lies in the functioning of the STAC anti-collision protocol. In STAC, when a reader does not receive any tag response during a slot, it sends the "end slot" command earlier than the slot in which it receives a response. This means that the size of an "empty" slot, i.e, a slot in which the reader cannot successfully decode a tag response, is smaller than a slot in which tag response is heard successfully. Thus, since the random and naive protocols are able to read fewer tags successfully, due to reader-tag or reader-reader collisions, they complete the reads faster than the CSMA protocol.

We will now show results for three random configurations. These configurations with the location of the RFIDMotes in the 2D plane and their conflict graphs are shown in Figures 7(a) and 7(b), respectively.

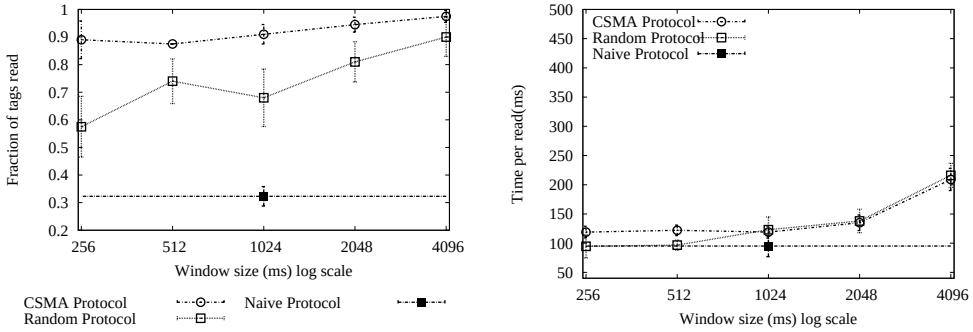
The performance results are shown in Figures 7–9 for these three configurations. Note that the configurations A



(a) Random configurations with locations of RFID-Motes shown. The scale is in cm.

(b) Conflict graphs for the random configurations.

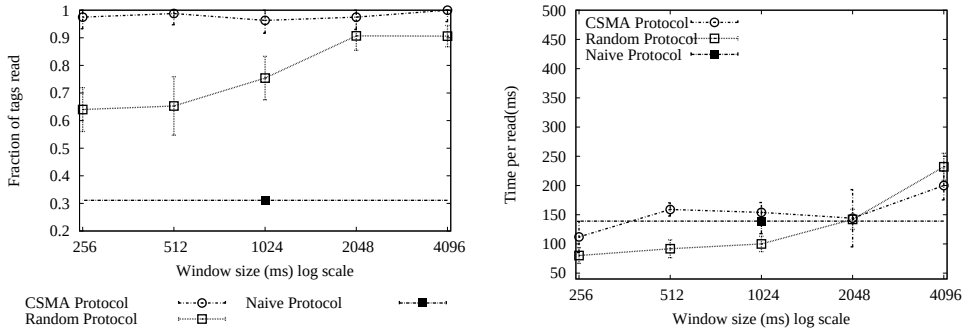
Figure 7: Random configurations and their conflict graphs.



(a) Accuracy.

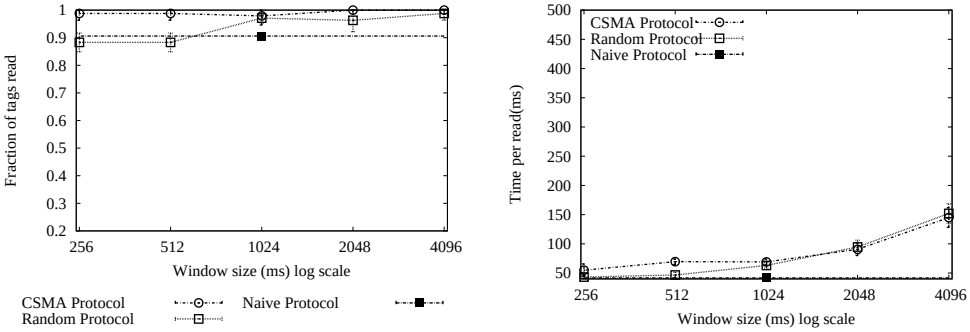
(b) Time per read.

Figure 8: Accuracy and time per read vs. window size for the scenario in Figure 7(a)A.



(a) Accuracy.

Figure 9: Accuracy and time per read vs. window size for the scenario in Figure 7(a)B.



(a) Accuracy.

(b) Time taken per read.

Figure 10: Accuracy and time per read vs. window size for the scenario in Figure 7(a)C.

and B have several conflicts and C has none. Thus, as expected CSMA provides much superior performance in configurations A and B and the naive protocol performs the worst. The protocols perform almost similarly in configuration C due to the absence of conflicts. But still CSMA has a slight advantage because it appears that occasional stray signals still cause a few collisions in the other two protocols.

Finally, note that 95% confidence interval for the CSMA has been usually much smaller than the random protocol. Thus, the performance of the CSMA protocol is more predictable.

6. CONCLUSION

In this paper, we have developed a CSMA-based MAC protocol to address reader-reader and reader-tag collision problems in RFID networks. In order to realize this protocol in a working system, we have built the carrier sensing capability in a commercially available HF RFID reader OEM module and implemented the MAC protocol on the reader. We have created topologies that may represent actual deployment scenarios and ran some experiments to analyze the performance of the protocol. We have shown that the protocol is indeed able to achieve superior performance relative to other alternatives that do not rely on carrier sensing. While carrier sensing is an established technique for multiple access and is indeed expected to perform very well, our work demonstrates the feasibility of using carrier-sensing as an add-on at a low cost for tiny HF readers that otherwise have not been developed for multi-reader environments.

We are currently in the process of augmenting our testbed to a larger number of RFIDMotes and evaluating performance in more varied deployment scenarios.

7. REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification.* New York, NY, USA: John Wiley & Sons, Inc., 2003.

[2] "Smart Dust, Project Home Page, http://robotics.eecs.berkeley.edu/ pister/SmartDust."

[3] S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID Systems and Security and Privacy Implications," in *Workshop on Cryptographic Hardware and Embedded Systems*, ser. Lecture Notes in Computer Science, vol. 2523, 2002, pp. 454–470.

[4] "3.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interference Specification: Candidate recommendation, Version 1.0.0." Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2003., AutoID Center, Tech. Rep.

[5] "EPC Generation 1 Tag Data Standards Version 1.1 Rev.1.27," EPC Global, Tech. Rep., May 2005.

[6] "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.0.9," EPC Global, Tech. Rep., January 2005.

[7] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision Problem," in *IEEE International Conference on Communications*, vol. 2, 2002, pp. 1206–1210.

[8] V. Deolalikar, M. Mesarina, J. Recker, D. Das, and S. Pradhan, "Perturbative Time and Frequency Allocations for RFID Reader Networks," in *HP Lab Technical Report*, 2005.

[9] "Skyetek,inc. http://www.skyetek.com."

[10] "Crossbow Technologies,Inc. http://www.xbow.com."

[11] "TinyOS Community Forum http://www.tinyos.net."

[12] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in *Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93–104.

[13] Literature Number 11-08-26-003, *HF Antenna Design Notes Technical Application Report*, 3rd ed., Texas Instruments, September 2002.

[14] *5 MHz-400 MHx 100dB High Precision Limiting-Logarithmic Amplifier AD8306*, Analog Devices.