New Reader Anti-collision Algorithm for Dense RFID Environments

Christine Meguerditchian, Haidar Safa, and Wassim El-Hajj

Department of Computer Science
American University of Beirut
Beirut, Lebanon
Email: {ckm03, hs33, we07}@aub.edu.lb

Abstract— **In dense Radio Frequency Identification environments, several readers are placed in the same area to scan a large number of tags covering a wide distance range. The placement of the RFID elements may result in several types of collisions. This paper proposes a new distributed multi-channel algorithm to solve the reader collision problems in dense RFID environments. The proposed algorithm aims at minimizing the identification delay, collision probabilities, and network overheads. We have evaluated the performance of the proposed approach and compared it to several reader collision solutions found in the literature such as NFRA, Dica and McMac. The results show that the proposed approach reduces reader collisions while minimizing the total interrogation time and the network overheads.**

I. Introduction

As part of the "Automatic Identification and Data Capture" group, the Radio Frequency Identification (RFID) started in 1973 to replace the traditional use of bar codes, especially in the retail market and supply chains. Unlike barcodes, RFID enables the wireless interaction over certain frequencies of RFID readers with a network system, to uniquely identify, track and capture the status of tagged objects within packages, animals or people at varying distances without the need of human intervention. These unique properties simulated an increased interest in deploying RFID systems in various fields, such as health, sports, warehouse inventories and object tracking.

An RFID network is composed of four main elements: 1) RFID tags, 2) RFID readers, 3) the air interface, and 4) edge servers. Typically, RFID readers emit radio-frequency signals that RFID tags would detect if present in the reader's transmission range. RFID tags respond to the reader's queries by emitting radio waves back with the data stored in the chip.

In dense RFID environments, several readers are placed in the same area to scan a large number of tags (Fig. 1)for a desired coverage range which may cause collisions. Collisions reduce the throughput of data collection, increase the identification delay, and degrade the system's efficiency and reliability. Three types of RFID collisions exist: 1) tag to tag collisions, 2) reader to reader interference (RRI), and 3) reader to tag interference (RTI). A tag to tag collision occurs when a reader tries to read multiple tags simultaneously. Many efforts have been done in the literature [1] to minimize this type of collisions, however the solutions for this problem are limited since most of the practical deployments use passive low cost tags, which cannot be programmed and cannot participate effectively in the anti-collision algorithms. In this paper, we focus on the reader collisions, RRI and RTI.

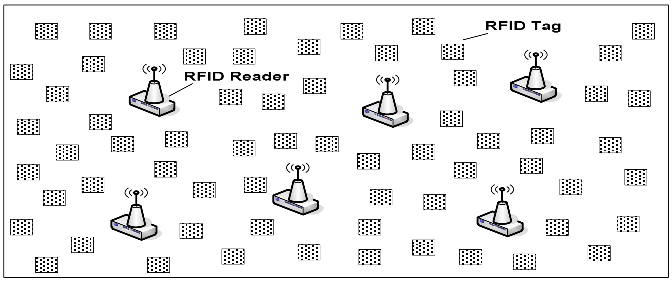


Figure 1. An example of a dense RFID environment

To illustrate the reader collision problem, it is essential to differentiate between the transmission range of a reader and its interference range as shown in Fig. 2. This figure contains two readers, R1 and R2, and two tags, T1 and T2. The transmission/read range is the coverage area of the reader, which reaches 10 meters when the reader is operating with an output power of 2W [2], while the interference range is the area that the reader causes interference on, which may reach 1000 meters [3]. RRI occurs when R1 attempts to read data from T1 using a channel with frequency f_1 and R2 is trying to read data from tags in its transmission range (example T2) using the same channel with frequency f_1. The signal sent from R2 to read T2 will interfere with the reply signal sent from T1 to R1. RRI can be avoided by having the readers operate at different frequencies or different time slots [1].

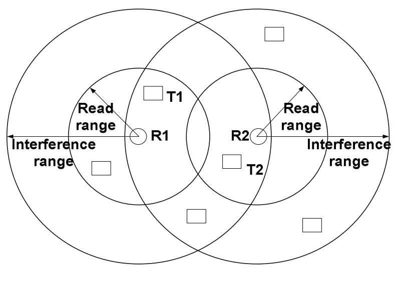
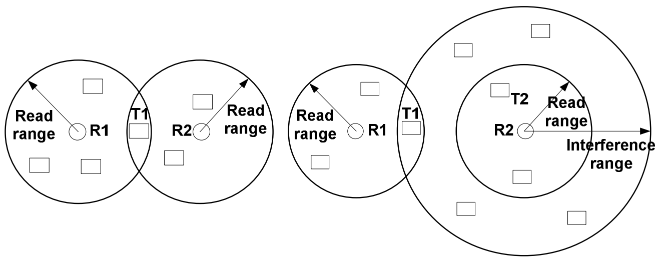


Figure 2. RRI Collision

On the other hand, two types of RTI exist. The first type is shown in Fig. 3 (a) and occurs when R1 and R2 attempt to read T1 simultaneously. The tag T1 will not be able to decode the commands of both readers and consequently will not be able to

978-1-4577-1846-5/11/$26.00 ©2011 IEEE

reply. This type of collision can be avoided by having the readers operate at different time slots [1]. The second type of RTI is shown in Fig. 3 (b) and occurs when R1 and R2 attempts simultaneously to read T1 and T2 respectively using frequency f_1. Since T1 is in the interference range of R2 and the read range of R1, both signals will reach T1 and collision will occur at the tag. This type of collision can be avoided by having the readers operate at different frequencies or at different time slots.



(a) (b)

Figure 3. (a) first type of RTI (b) second type of RTI

The remainder of this paper is organized as follows. In section II, we survey some related work. In section III, we present our new approach. In section IV, we evaluate the performance of the proposed approach and compare it with other algorithms. Conclusion is presented in section V.

II. Related Work

Many approaches were recently proposed to reduce the impact of collisions, minimize interference, and maximize the read range [5-10]. A distributed solution for collision avoidance (Dica) was proposed in [4]. In Dica, each reader repetitively contends with other readers through a control channel. The winner of the contention can start using the data channel for tag interrogation while the loser must wait till the channel is idle. In this approach, a reader sends a "BRD_WHO" packet to identify operating readers in its vicinity. A "BUSY" packet is sent as a reply to the "BRD_WHO" message to notify that a reader is currently reading tags. A "BRD_END" packet is sent to notify neighboring readers that the interrogation is finished and the channel is idle. To avoid hidden and exposed terminal problems the range of the control channel is set to double the transmission range of a reader. Nevertheless, Dica suffers from ignoring RRI collisions and uses only one data channel.

A multi-channel MAC protocol for RFID networks (MCMAC) was proposed in [5]. MCMAC also uses a control channel to exchange control packets between readers, but unlike Dica, its communication range is set such that, any two readers that can interfere with each other on the data channel, are able to communicate on the control channel. It distributes the data channels among the readers using a random access algorithm. It defines three stages a reader must perform before starting the interrogation. During the listening stage, a reader must listen to the control channel for $T_{min}+ i$ time, where i belongs to interval [0,..,CW]. If it receives a control message, it analyzes the message to find which channel has been occupied and whether there is an idle channel to utilize. If there are idle channels, the reader selects one and sends out a control message to denote that the channel has been occupied. If there are no idle channels to use, the reader loses the cycle and waits for the next cycle. During the transmission control stage, the reader sends a control packet to notify others of the selected channel. During the reading stage, the reader keeps sending periodically control packets to neighboring readers to notify them that it is still using the channel for reading tags. This approach completely solves RRI and the second type of RTI, but does not address the first type of RTI. Therefore, two readers can be using different data channels, but still cause collisions if they are reading the same tag simultaneously. Also, MCMAC doesn't address collision between control packets.

A neighbor friendly anti-collision solution (NFRA) was proposed in [6]. It solves both RRI and RTI through a centralized algorithm for fixed and mobile readers. A polling server broadcasts an arrangement command (AC) which includes the range of random numbers. The readers that receive the AC, generate their own random numbers. Then the server issues an ordering command (OC). Each reader compares its random number with the value in the OC. If they are the same, the reader issues a beacon to determine whether a collision occurs or not. After the beacon frames, if some readers do not detect any collisions, they send overriding frame (OF) to the neighboring readers. The OF prevents the neighboring readers from receiving the next OC from the server. The neighboring readers which do not identify the next OC due to the OF or which detect a collision of beacons do not conduct identification of tags until the next AC. NFRA, like Dica, assumes the use of only one data channel, and it doesn't mention how the collision between the beacons is detected by the readers.

We compare in table 1 the characteristics of the approaches discussed above along with other solutions proposed in literature to solve RRI and/or RTI. The approaches can be classifies as centralized or distributed, some of which use a single data channel, while others use multiple ones. Our proposed anti-collision algorithm makes use of multiple data channels for the communication between the readers and the tags and two control channels for a notification mechanism. It is distributed, hence avoiding the need of extra costly hardware for centralized control. It is also suitable for mobile readers which are more and more deployed for dynamic data capture replacing the fixed static readers.

TABLE I. Dense RFID Anti-collision Solutions

	PULSE [7]	DAPC [8]	MCMAC [5]	Dica [4]	RAMP [9]	NFRA [6]	New
RRI	X		X		X	X	X
RTI	X	X		X		X	X
Centralized		X				X	
Distributed	X	X	X	X	X		X
Multiple data channels			X		X		X
Use of control channel	X		X	X	X	X	X
Suitable for mobile readers	X		X	X	X	X	X
FDMA			X				X
TDMA			X			X	X
CSMA	X		X	X	X	X	X

III. Proposed Appraoch

The proposed algorithm builds a notification mechanism between the readers to make them aware of the resources, channels and time allocations, being used in the network. To describe the proposed approach, we first define the following notations. For every reader R_i we define the read range as r_i, the interference range as I_i and the distance between reader i and every other reader j as $D_{i,j}$.

For two readers R_i and R_j:

- RTI occurs when: $D_{i,j} < r_i + r_j$ and in this case the readers must operate at different time slots to avoid the collision.
- both RRI and RTI occur when: $D_{i,j} < r_i + I_j$ and in this case the readers must operate at different frequencies to avoid the collision.

Each reader will have two queues: 1) Queue 1, Q1, which stores packets received from readers that interfere with its operation if the same frequency is used, 2) Queue 2, Q2, which stores packets received from readers that interfere with its operation if they operate simultaneously. Two control channels will be used for the notification mechanism between the readers: control channel 1 (CC1), which has a transmission range equal to $r + I$, and control channel 2 (CC2), which has transmission range equal to $2r$. The number of initially available data channels is n. The proposed algorithm is composed of two parts, the reader operation part (i.e., Contention and interrogation) and the reader receiving part as shown in Fig. 4. Two types of messages are exchanged: *Start* packet and *End* packet.

```
Subroutine1: Contention and interrogation
1.  Available channels AV = {c1,…,cn}
2.  Wait for pseudo-random time
3.  If size(set(Q1)) != n and size(Q2) = 0
4.      Choose a channel from {channels(AV) – channels(Q1)}
5.      Send a Start packet on CC1
6.      Send a Start packet on CC2
7.      Start tag interrogation
8.  Else
9.      Wait for a pseudo-random time and go back to 3
10. If tag interrogation is over
11.     Send End packet on CC1
12.     Send End packet on CC2

Subroutine2: Receive
1.  If Type(ReceivedPacket)=start on CC1
2.      Add ReceivedPacket to Q1
3.  Else
4.      If Type(ReceivedPacket)=start on CC2
5.          Add ReceivedPacket to Q2
6.      Else
7.          If Type(ReceivedPacket)=end on CC1
8.              Remove the packet having the same ReaderId from Q1
9.          Else
10.             If Type(ReceivedPacket)=end on CC2
11.                 Remove the packet having the same ReaderId
```

Figure 4. Pseudocode of the proposed approach

When a reader wants to start an interrogation, it starts by checking both its Q1 and Q2. If the size of the set of Q1 is not equal to n (i.e., not all the data channels are being used by the neighbor readers that interfere with it on the data channel), and if Q2 is empty (i.e., none of the neighboring readers, operating simultaneously, that interfere with it is currently interrogating tags), it sends a *Start* packet on CC1 and CC2 and starts the tag interrogation. If one of the two conditions is not satisfied, the reader waits for a pseudo-random time and tries again. During the contention and tag interrogation phase, if the reader receives any type of control packets, it queues them in the corresponding queue depending on their type and the control channel over which it received them as shown in subroutine 2 of Fig. 4. The idea behind setting the transmission range of CC1 to $r + I$ is to notify the neighbor readers that cause RRI or the second type of RTI with the interrogating reader that it is using a certain data channel, hence they choose a different data channel if any is available (line 4 of subroutine1 of Fig. 2). On the other hand, the transmission range of CC2 is set to *2r* to notify the neighbor readers that cause the first type of RTI with the interrogating reader that the reader is operating, hence they cannot operate simultaneously by condition 2 of line 3 of subroutine1 of Fig. 4. When the reader finishes interrogating tags, it sends *End* packets on both CC1 and CC2 to notify them that it is over, and the receiving readers remove the corresponding *Start* packets from their queues.

The packet format used for the control messages is shown in Fig. 5. For the *End* packets, the *Selected Channel* field is left empty, because to remove a *Start* packet from the queue, we only need its type and the Reader ID to identify it.

Type	Reader ID	Selected Channel

Figure 5. Packet format

IV. Simulation Results

In this section, we study the performance of our approach and compare it with the following three algorithms: Dica, Mcmac and NFRA. All the algorithms are implemented using the network simulator NS3. We simulated the RFID network using the RangePropagationLossModel and the 802.11 Wifi model of NS3 with the RTS/CTS and SIFS/DIFS disabled on the MAC layer like done in NS2 in [10]. In this study, 25 readers are deployed at varying distances from each other. The transmission range of the readers is 10 meters while the interference range is 1000 meters. The number of available data channels n is set to 6.

Fig. 6 shows that the number of failed interrogations increases as the number of readers increases in both Dica and Mcmac. That is because Dica solves RTI only, while Mcmac solves RRI only. On the other hand, using our approach, no failed interrogations are encountered for any number of readers, just like in NFRA, which also solves RTI and RRI however using a centralized approach.

Fig. 7 shows how our approach improves the total interrogation time for all readers in the network. We assume that each reader has to identify 100 tags in its range, where the time needed is 0.46 seconds [6]. Since Dica and Mcmac have a big number of collisions in the network, they have low interrogation time for any number of readers. While both our approach and NFRA fully solve collisions, our proposed algorithm yields a much better interrogation delay for any number of readers in the network and increases linearly as the number of readers increases in contrast to the NFRA, where the delay increases exponentially when the number of readers exceeds
10.

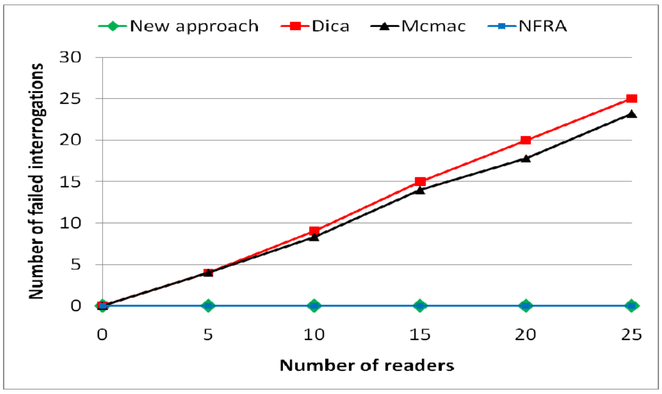


Figure 6. Number of failed interrogations versus the number of readers

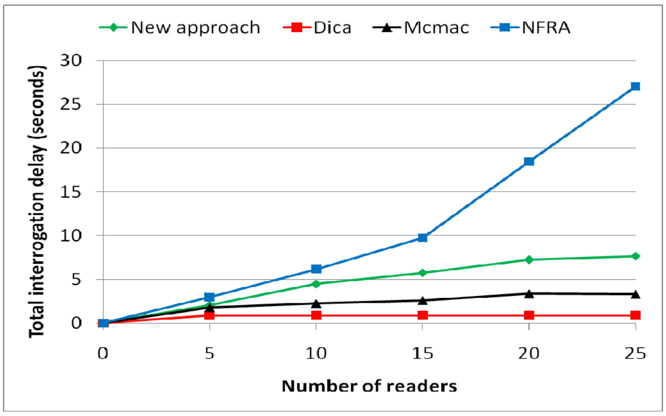


Figure 7. Total interrogation time versus the number of readers

Fig. 8 shows the amount of the network overhead of the four algorithms. To measure overheads, we count all the control packets exchanged between the readers, or between the readers and the polling server, whether during the contention phase or the interrogation phase itself. The figure shows that our approach has the lowest overheads in the network after Dica, which doesn't solve all the collisions, and Mcmac has the highest overheads. Compared to NFRA which solves all the reader collisions, our approach has a lower number of overheads in the network, especially when the number of readers exceeds 15.

V. CONCLUSION

In this paper, we proposed a new distributed multi-channel algorithm that solves both RTI and RRI reader collisions using a notification mechanism that is used to make RFID readers aware of the utilization of the networks resources. We evaluate the proposed approach through simulations using a varying number of readers. The simulation results show that our approach solves both types of reader collisions while minimizing the total interrogation time and the overheads in the network.

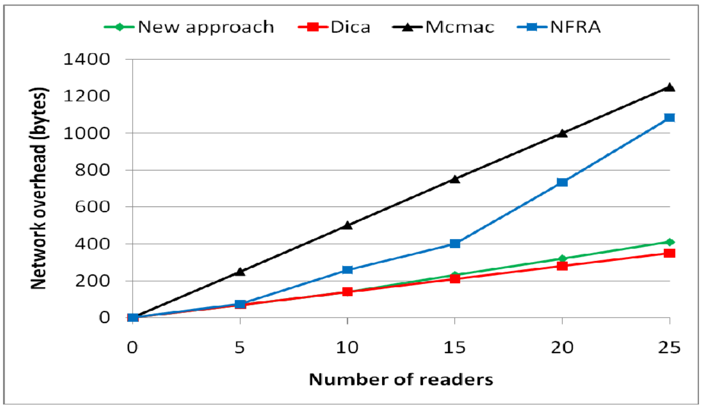


Figure 8. Network overhead versus the number of readers

REFERENCES

[1] Klair, D.K.; Kwan-Wu Chin; Raad, R.; "A Survey and Tutorial of RFID Anti-Collision Protocols," Communications Surveys & Tutorials, IEEE , vol.12, no.3, pp.400-421, Third Quarter 2010 doi: 10.1109/SURV.2010.031810.00037

[2] ETSI-EN 302 208-2 v1.1.1, September 2004. CTAN. Available online at: http://www.etsi.org

[3] K.S. Leong, M.L. Ng, P.H. Cole, "The reader collision problem in RFID systems", in Proc. of IEEE International Symposium on Microwave,Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings, pp. 658-661, 2005.

[4] Kwang-il Hwang; Sang-Soo Yeo; Jong Hyuk Park; , "Distributed Tag Access with Collision-Avoidance among Mobile RFID Readers," CSE '09. International Conference on Computational Science and Engineering, 2009., vol.2, no., pp.621-626, 29-31 Aug. 2009 doi: 10.1109/CSE.2009.342

[5] Dai, H., Lai, S., Zhu, H."A Multi-Channel MAC Protocol for RFID Reader Networks" WiCom 2007, 21-25 pp: 2093-2096. 2007.

[6] J-B. Eom, S-B. Yim, T-J. Lee, "An Efficient Reader Anticollision Algorithm in Dense RFID Networks with Mobile RFID Readers", IEEETransactions on Industrial Electronics, Vol. 56, no 7, pp.2326-2336, July 2009

[7] Birari, S. M., and Iyer, S. "PULSE: A MAC Protocol for RFID Networks" 1st International Workshop on RFID and Ubiquitous Sensor Networks, Japan, Dec 2005.

[8] K. Cha, S. Jagannathan," Adaptive Power Control Protocol With Hardware Implementation for Wireless Sensor and RFID Reader Networks", IEEE SYSTEMS JOURNAL, VOL. 1, NO. 2, DECEMBER 2007, pp. 145 v- 159

[9] G. Joshi, K. Mamun, S. Kim, "A Reader Anti-collision MAC Protocol for Dense Reader RFID System", 2009 International Conference on Communications and Mobile Computing, pp. 313-316.

[10] Arini Balakrishnan, Swetha Krushnan, "Simulation of RFID platform on NS-2", IEEE, 2005, pp. 1-12.