Journal of Ambient Intelligence and Humanized Computing (2019) 10:4043–4057
https://doi.org/10.1007/s12652-019-01221-5

ORIGINAL RESEARCH

Scheduling RFID networks in the IoT and smart health era

Fabio Campioni[1] · Salimur Choudhury[2] · Fadi Al- Turjman[3]

Received: 28 March 2018 / Accepted: 21 January 2019 / Published online: 31 January 2019

Abstract

As a potential way to dramatically save energy and live in a green and smarter planet, the internet of things (IoT) aims to utilize energy-efficient enabling technologies such as the RFID systems in our daily life applications. RFID, or Radio Frequency Identification, is used to efficiently locate items using tags and readers. In this paper, we propose localized reader scheduling algorithms for RFID networks. We consider readers with limited amounts of energy, powered by a battery. Using only local information, the readers schedule themselves to minimize energy usage and maximize network lifetime. We compare the performance of our localized algorithms to a centralized heuristic (the research problem is NP hard) based on a set cover approximation solution and show that the localized algorithms obtain equal or better performance in comparison to centralized solution, achieving 5% higher area under the curve (AUC) in scenarios with 50% readers, and 13 and 8% higher AUC in 25% and 15% reader scenarios, respectively.

Keywords RFID Networks · Scheduling · IoT · Localized algorithms

1 Introduction

Internet of things (IoT) introduced a radical evolution of the Internet as it was known, into a ubiquitous network of interconnected objects that are capable of harvesting information from the environments via sensors and RFID (Radio Frequency Identification) systems. It interacts with the physical world and at the same time uses existing Internet standards to provide services for information transfer, analytics, and applications. RFID networks play an important role in the area of smart health care (Natarajan et al. 2016). RFID is used to efficiently track physical items using radio communication. RFID networks consist of two components: readers and tags (Domdouzis et al. 2007). Readers use radio frequencies to detect nearby tags. Tags are attached to items being tracked and can be active or passive. They contain memory that can store a unique identifier, as well as any additional information required by the application (Lozano-Nieto 2013). Active tags have a battery and can typically switch between radio frequencies; passive tags have no battery and instead use energy harvested from the radio interrogation signals transmitted by readers. RFID systems with passive tags have shorter ranges than active tags, so more readers have to be placed and need to be deployed close to each other to maximize tag coverage. However, passive tags are much cheaper than active tags so are therefore more commonly found in real world applications.

There are several benefits to using RFID. Since antennas used in RFID networks can be bidirectional, configuration is simplified as no external sensor is needed. When using passive tags, no additional energy source is required as the tag harvests energy from received radio signals. Depending on the frequency and antenna gain, medium ranges are easily attainable from 10 meters to up to 100 meters. The cost of passive tags is also an advantage to RFID; typical passive tags cost only 10 to 20 cents each. Many tags are also flexible, allowing adhesion to nearly any physical object. This makes them ideal for systems that wish to track physical items at a low cost (Domdouzis et al. 2007).

✉ Salimur Choudhury
salimur.choudhury@lakeheadu.ca

Fabio Campioni
campioni@cs.queensu.ca

Fadi Al- Turjman
fadi@metu.edu.tr

[1] School of Computing, Queen's University, Kingston, ON, Canada

[2] Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada

[3] Department of Computer Engineering, Antalya Bilim University, Antalya, Turkey

Dense placement of readers results in a dense reader environment, which suffers from interference and collision issues due to overlapping reader interrogation zones. A reader interrogation zone is the area around a reader in which it can communicate with tags. Collisions prevent tag replies from reaching the readers, which prevents items from being tracked correctly. Many algorithms exist for reducing or eliminating collisions, by either scheduling reader transmissions or eliminating redundant readers (Wang and Liu 2017; Rashid et al. 2018; Chen et al. 2009). Two main types of collisions occur in RFID networks. The first type, reader to tag, occurs when one tag is located in the interrogation zone of two or more readers at the same time. If more than one reader tries to interrogate that tag at the same time, the transmissions will not be successful. With reader to reader collisions, signals emitted from one reader prevent another reader from successfully communicating with other tags (Chen et al. 2009). Many algorithms have been developed to alleviate these issues, several of which will be explained in the related work section. Our main contributions in this paper can be summarized as follows. An RFID simulator, based on a cellular automata model, is designed and implemented. A number of RFID readers' scheduling algorithms are developed:

- Centralized algorithm based on greedy set cover approximation that places readers in disjoint sets.
- Localized algorithm utilizing indirect communication via tag state.
- Localized algorithm 2, utilizing direct reader communication.
- Localized algorithm 3, utilizing direct reader communication with forced sleep probability.

Extensive performance evaluations show that the first localized algorithm achieves similar performance to the centralized algorithm, and in some cases, particularly scenarios with extremely sparse reader placements, higher overall tag coverage is achieved in comparison to the centralized algorithm.

In Sect. 2, we will discuss some related work including existing algorithms for related problems. In Sect. 3, we describe the cellular automata model taken into consideration when designing and implementing our algorithms. In Sect. 4, we describe the transition rules for each of our proposed algorithms. In Sect. 5, we implement and test our algorithms on a variety of randomly generated scenarios, and compare our localized algorithms with the centralized algorithms. Finally, we conclude our findings in Sect. 6.

2 Related work

Some previous work in RFID networks include redundant reader elimination and collision prevention. Redundant reader elimination involves detecting readers that are not necessary and disabling them. This is typically done during the setup of a network, rather than throughout its runtime. Collision prevention involves scheduling reader transmissions in time slots so that adjacent readers do not transmit at the same time and cause collisions.

The localized algorithms presented in this paper are inspired from the algorithms found in Rashid et al. (2016), Rashid et al. (2018). In the paper, the authors model an RFID network as a cellular automaton. Each reader writes its state into neighboring tags and then checks to see if it is the only reader covering each tag. If any such tag is found, the reader will permanently remain awake for the rest of the network lifetime, and writes its new state into each of its neighboring tags. Other readers still in the checking state check to see if all of its neighboring tags are covered by at least one reader in the awake state. If so, the reader will go to sleep and update its state in neighboring tags. Otherwise, it counts the number of awake and checking readers. If this number is less than k, the reader will plan to go to sleep. At this point, each tag checks each of its readers. If it finds that all are planning to go to sleep, it will randomly choose one to stay awake by writing into that reader's state. Each reader then again checks its neighboring tags states. If it finds that the tag has changed its value, it will remain awake. Otherwise, it goes to sleep. This approach ensures that a tag is always covered. This algorithm is designed to detect and disable redundant readers in an RFID network. A similar approach can be taken for power management of readers, which is why the localized algorithm presented in this paper uses concepts from this algorithm.

While the algorithm in Rashid et al. (2016) prevents collisions by detecting and eliminating redundant readers, the algorithm presented in Waldrop et al. (2003), called Colorwave, schedules reader transmissions so that they do not occur at the same time. The authors develop two distributed algorithms for preventing collisions in RFID networks. They approach it as a graph colouring problem, where readers are vertices in a graph and potential collisions between readers are edges in the graph. Each reader operates with only local information; no global sharing of information is required. This allows the algorithm to adapt to disturbances such as new readers entering the network hand held readers moving along a network. The goal of the algorithms is to color each node such that there is a minimal number of adjacent nodes of the same color. The first algorithm, known as distributed color selection (DCS), takes one input, which is the maximum number of colors. Each reader chooses a random color that corresponds to a time slot, and can only transmit during that time slot. If a transmission happens to collide with another reader, the transmission request is discarded, and the reader randomly chooses a new color. All neighbors of that reader also randomly choose a color. Additionally, each reader keeps track of what color it believes the current time slot to be. Because the algorithm is distributed, the time slot ID does not need to be synchronized. This algorithm was also considered as a starting point as it relied on only local information, while obtaining near optimal performance.

The other algorithm described in Waldrop et al. (2003) is referred to as variable maximum distributed color selection (VDCS), or Colorwave. It is an extension of DCS, but with 5 input values rather than one. Each reader monitors the percentage of succesful transmissions, which are compared to the inputs to make decisions. For example, when a reader reaches a safe percentage to change its own maximum number of colors, it sends a kick to neighbours to do the same. Because of the higher number of inputs, Colorwave is more flexible than DCS and existing linear backoff algorithms as well. The results prove that the algorithm performs better than the static DCS algorithm and existing linear backoff algorithms. While the two algorithms presented in the paper achieve good performance using local information, power management is a different problem than transmission scheduling.

Another algorithm that attempts to build on the ideas of the authors of Colorwave (Waldrop et al. 2003) is MALICO (Bueno-Delgado and Pavón-Mariño 2013). It is based on Colorwave, but makes a few changes to make it more flexible. It is more adaptable to changing scenarios, including mobile readers leaving and entering the network. Instead of limiting readers to operating on one frequency, it allows readers to operate on up to four frequencies. Readers are assumed to have two bi-directional antennas, so that they may read and write at the same time. They use this capability to be able to listen at the same time that they are transmitting. Upon entering a network, a reader chooses one of the four frequencies to operate on. It starts the identification round with a random number of colours. A reader selects a random colour and starts listening for a variety of information, including collisions and successful transmissions. When the selected color starts, the reader tries to communicate with its tags. If a collision occurs, the reader stops transmitting and keeps listening until the end of the round. At the end of the identification round, it uses the data it collected to estimate the number of readers in its working frequency. Once the number of readers have been estimated, the length of the next identification round is adjusted to the new number of colors, and the next identification round begins. Again, scheduling transmissions is a different problem than power management, so these concepts couldn't really be used in the algorithms found in this paper.

OSL, described in Vales-Alonso et al. (2016), describes a centralized scheduling algorithm for dense RFID networks. The authors show that a centralized node can assign resources in such a way to reduce collisions. First, the topology of deployed readers and tags is collected by the central node for a preprocessing phase. In addition, the number of tags each reader covers, referred to as load, is also observed. The scheduler uses this information to generate a schedule that maximizes network throughput (number of tags being read) while minimizing collisions. The scheduler uses a simulated annealing solver, which tries random variations of the current solution, accepting new solutions if they provide increases throughput. There is no guarantee that the final solution will be the best solution, but finds a solution that is good enough.

In Cardei and Du (2005), several approaches are described for extending operational time within wireless sensor networks. It does this by organizing sensors into a maximal number of disjoint set covers that are activated successively. Each set must completely cover all targets. Then, the algorithm cycles through all subsets. The current subset is considered active, while the others are put into a low energy usage sleep state. By determining the maximum number of disjoint sets, the time interval between sets is longer, resulting in a network that lasts longer overall. The algorithm runs on a central node after the location of all targets is known. Once the sets have been determined, the central node lets each sensor know what set it is in. The sensor then knows when it has to be active or asleep. The algorithm is simply an application of the disjoint set cover problem, and the authors demonstrate that it is an NP complete problem. While this algorithm was implemented for wireless sensor networks, it could be easily applied to RFID networks.

In Yang et al. (2017), the authors propose the M-ICHB algorithm and a set of protocols based on it for homogeneous and hetergeneous wireless sensor networks. The algorithm and protocols aim to provide energy efficient clustering protocols. They compare their protocols to existing protocols and find that they improve the stability region 53 to 58%, and improve the number of received packets at the base station by 91 to 97% compared to the existing LEACH protocol.

In El Fissaoui et al. (2018), the authors propose an energy efficient, fault tolerant, distributed algorithm for data aggregation in wireless sensor networks. They adopt a clustering method to group sensor nodes into clusters, and then plan efficient itineraries among cluster heads in the network, by calculating the impact factor of each node in the network and selecting the nodes with the highest impact factor as cluster nodes. Then, a mobile agent is dispatched to collect data from all of the cluster head nodes. In their performance evaluations, they perform simulations which show that their approach significantly improves network lifetime, and has consistently shorter itineraries compared to existing approaches.

In Lounis et al. (2015), the authors develop a parallel, GPU based simulator for accurate simulation of wireless sensor networks. They demonstrate it by simulating energy consumption of wireless sensor nodes. They compare their approach to existing sequential simulation models. By exploiting the highly parallel, SIMD architecture of modern GPUs, they find that they achieve up to 25× faster simulation times compared to sequential CPU based simulation models. If processing on each sensor node is more time consuming, the accelerations in simulation time are even better. However, they find that in simulations with 500 sensor nodes or less, they find no speedup between CPU and GPU based simulations.

In Yang et al. (2017), the authors propose two new accurate fitness functions for clustering and routing, as well as new greedy particle swarm optimization protocol for clustering and routing in wireless sensor networks. They propose a eGMDPSO-C clustering algorithm as well as a eGMDPSO-R routing algorithm. They compare their algorithms to LEACH, SEP, ERP, and TPSO-CR in a series of simulations, which show that the proposed protocol prolongs network lifetime, increases network coverage, improves network throughput, and maintains acceptable energy consumption in comparison to existing approaches.

Kumar and Kumar (2018a) propose a load balancing method that overcomes the nonuniform load problem which causes uneven energy usage in wireless sensor networks. It ensures full coverage and connectivity with the least possible number of nodes. Their approach controls energy disparity by clustering the network and providing a set of communication and sensing ranges. They evaluate their approach based on network lifetime and coverage using simulations, which show that the network lifetime is improved by up to 17% compared to existing approaches.

Kumar and Kumar (2018b) also propose LARCMS, an algorithm which updates the position of mobile sinks in wireless sensor networks to alleviate the issue where sensor nodes closer to sink nodes exhaust their energy faster compared to sensor nodes farther away. Location aware nodes are used to store updated positions of mobile sinks, and normal nodes contact these location aware nodes to get the location of mobile sink nodes. They then forward data to the mobile sinks via geographic forwarding. They compare their technique to existing LBDD, railroad, and ring routing approaches in terms of average energy consumption per node, network lifetime, and average reporting delay. In their simulations, they show that their approach enhances network lifetime by 1.54 times and reduces reporting delay by 1.57 times.

We did not compare to existing algorithms because they solve different problems. Our algorithms are designed specifically for the energy-based reader scheduling problem, which is scheduling mobile, battery powered readers after they have already been deployed and switching states for the goal of saving energy. These existing algorithms either coordinate communication between readers in order to reduce collisions, or eliminate redundant readers before they are deployed (during the network setup phase) (Table 1).

3 Cellular automata-based algorithms

The research question of this paper asks whether a localized RFID scheduling algorithm can be developed that achieves comparable results to a centralized scheduling algorithm. By running scenarios in a simulator, algorithms can be implemented and compared quickly. Results are output to a graph, and the various algorithms are compared to determine which provides the best tag coverage and network lifetime among varying topologies, and how different variables affect behavior of the readers.

The localized algorithms are simulated using a cellular automata model. A cellular automata is biologically inspired model commonly used to model physical systems (Choudhury et al. 2012c), (Choudhury et al. 2012a), (Choudhury et al. 2014), (Choudhury et al. 2012b), (Choudhury 2012), (Hassan et al. 2018). Since our algorithms draw inspiration from the ones proposed by Rashid et. al in (Rashid et al. 2016), which also use a cellular automata model, we chose to implement our algorithms using the same model. They consist of the following:

- C is a cellular grid
- S is the set of states
- δ is the transition rule of the automaton
- N is the neighbourhood of a cell

The grid C can be described as a Moore neighbourhood, as seen in Fig. 1. Each grid in the cell has 8 neighbours, represented by a cardinal direction. The state of a tag has one value for each direction around it. A reader can access a cell in any of the directions immediately around it (Rashid et al. 2016).

At each time t, each cell in C is assigned a state in S. The state of the cell at time $t + 1$ is determined by the transition function δ depending on the current state of c and the state of cells in its neighbourhood N (Garzon 2012).

In each simulation, a set of readers R and tags T are loaded from an input file containing their grid arrangement that is randomly generated. Each cell in the grid can be either a reader or a tag. A reader can communicate with any tags that are its immediate neighbour. A tag can also

Table 1 Comparison of existing works

Problem	Algorithm	Pros	Cons
Redundant Reader Elimination	CARRE	Eliminate redundant readers using only local information Easy to implement Does not require global identification of readers and tags Outperform all existing localized scheduling algorithms for redundant reader problem	Does not consider readers scheduling
Anti-Collision Schemes	Colorwave	Uses a graph colouring approach to schedule reader transmissions Propose two algorithms, DCS and VDCS, which support a fixed number and variable number of colours, respectively Results show better performance compared to existing backoff anticollision algorithms	Only supports readers operating on a single frequency channel
	MALICO	Extension of Colorwave- Allows scheduling of readers operating on multiple frequency channels Adapts to changeable identification scenarios by adapting the number of colors each round	Performance rapidly decreases when number of neighboring readers increases
	OSL	Uses simulated annealing solver to reduce collisions, maximize throughput Outperforms other centralized algorithms due to maximal efficiency and throughput maximization strategy	Centralized so requires a central node to perform scheduling

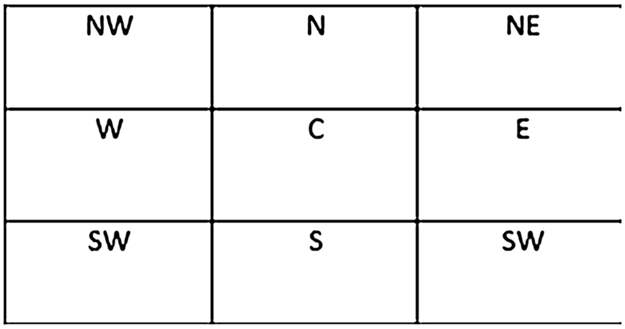


Fig. 1 Illustration of radius 1 Moore neighbourhood

be covered by any readers in its immediate neighbourhood, which means that a tag can be covered by more than one reader.

Given a grid C of a set of readers R and set of tags T, schedule the readers in such a way that maximizes both tag coverage and network lifetime. A tag is said to be covered if it is surrounded by at least one reader that is awake. When a reader is in the awake state, it loses energy and each reader has a limited amount of energy. Therefore, it is crucial that readers are efficiently scheduled so that only readers that need to be awake are, and others sleep until they are required. While a centralized algorithm has access to the location and information of all readers and tags in the network, localized algorithms can only access this information for cells directly around them. The challenge then is achieving similiar performance using only local information.

4 Proposed algorithms

4.1 Centralized algorithm

The centralized algorithm is a variation of the greedy set cover approximation problem (Chvatal 1979). However, the difference between the algorithm in Chvatal (1979) and the one mentioned in this paper is that this algorithm creates disjoint sets. This means that a reader can only be placed in one set for the entire network lifetime. Scheduling happens only once at the beginning of the simulation, compared to the localized algorithms, where scheduling happens on each time iteration. The centralized algorithm simulation does not use a cellular automata. Instead, the same scenario input is fed into a program that places all readers and tags into sets and creates disjoint subsets that are each run until all readers are exhausted of energy. Therefore, the algorithm steps are described rather than transition rules.

1. A set of readers R and tags T exist.
2. Add all tags in T to a set of uncovered tags.
3. For each reader $\in R$.

   Calculate tag cover of reader r. This is done by counting the number of its tags in T that are in the set of uncovered tags. That reader is removed from the set of readers R and placed in the current disjoint set.

The tags covered by that reader are temporarily removed from the set of tags T, and the tag cover of remaining readers is recalculated.

The reader with maximum is again removed from the set of readers and placed in the disjoint set. This process repeats until the remaining readers no longer cover any tags.

At that point, a new disjoint set is created, and the set of tags is restored to all tags in the simulation.

4. Once all readers have been placed into a set, the simulator loops through each set and activates readers on each time iteration until they are exhausted of energy.
5. Once a set is exhausted of energy, the simulator switches to the next set of readers.
6. This process repeats until there are no more remaining readers $\in R$.

This process repeats until all readers have been placed in a set. Then, the simulator begins simulating the network. The first set of readers runs until their energy is exhausted. Then, the next set is turned on. This repeats until all readers have been exhausted of energy.

4.1.1 Example

Using the same example scenario, the runtime of the centralized algorithm is described here. Readers are placed into a set, as well as all tags in the scenario. Reader 1 covers tags $\{T1\}$, reader 2 covers $\{T2\}$, reader 3 covers $\{T3\}$, reader 4 covers $\{T4\}$, and reader 5 covers $\{T1, T2, T3, T4\}$. The first disjoint set is created. The algorithm goes through each reader and calculates the number of tags covered. The reader covering the most tags is added to the set. Those tags are considered covered, and each reader is checked again to determine tag cover. In this case, reader 5 would be chosen, as it covers 4 tags while the others only cover 1. All tags covered by reader 5 are considered covered, and no more tags remain. The first disjoint set consists of only reader 5. Then, the algorithm repeats again with the remaining readers using a new disjoint set. Each reader has a tag cover of 1, so each is added to the new disjoint set. Since each tag is covered by only one reader, they are all added to the disjoint set. No other readers exist, so in this case, only two disjoint sets are created. The first set consisting of reader 5 will remain awake until exhausted of energy, and then the second set is activated until it runs out of energy.

4.2 Localized algorithm

All localized algorithms use the same set of states, S, so only their transition rules are described. The centralized algorithm does not utilize the cellular automata model, so no transition rule is described for that algorithm. It is instead a variation of the set cover approximation algorithm, that produces disjoint sets.

Each reader can be in one of three states: sleeping, checking, and awake. In the sleeping state, the readers consume little power and can not cover tags. In the sleeping state, there is a chance that the reader will wake up and switch to the checking state. In the checking state, readers observe their neighbours' state to determine how many tags they cover, and can determine energy levels of other readers by looking in tags' state. In the awake state, the readers cover tags. Readers only consume energy during the awake state, or when switching from the sleep state to the checking state.

The initial algorithm was inspired from the concepts presented in CARRE (Rashid et al. 2016). This algorithm had the issue where a reader may go to sleep even though it made more sense to stay awake, because it went to sleep after finding the first reader with more energy than itself, rather than taking all neighbours into consideration. For this reason, it did not achieve comparable results to the centralized solution, and results were not included in the performance evaluation.

4.2.1 Transition rules

(a) Each reader starts in the awake state.
(b) Each reader writes its energy, state, and current time into all of its neighbouring tags.
(c) It then checks each of its neighbouring tags state values. For each direction, if the cell is a reader, it checks the energy value of that reader and the time that it wrote the value. If the value is higher than this reader's energy, this reader will go to sleep. Otherwise, the reader stays awake.
(d) All readers will then be awake or sleeping.
(e) Any readers that are asleep randomly have a chance of waking up and going back into checking.

Each reader would begin by writing its energy into neighbouring tags' state. It would then check all neighbouring tag's state to see what other readers covered those tags. If it found a reader with higher energy than itself at any point, it would go to sleep. If it found a reader with the same amount of energy, it uses the reader's ID as the tiebreaker. Since all readers start with the same energy, reader 5 would remain awake, while the others go to sleep, and would cover all 4 tags. While asleep, there is a random chance that the other readers will wake back up to the checking state, where it would repeat the same process. Perhaps reader 4 decided to wake up and go back into checking. It would look at neighbouring tag states and see that it had more energy than reader 1. It would then switch into the awake state. Reader 5 would not immediately go to sleep until it randomly decided to switch to the checking state again and see that reader 4 had more energy than itself. This process repeats until all readers are exhausted of energy.

Another issue with this algorithm is that a reader will immediately go to sleep if it found a reader with more energy covering it, even if it made sense to stay awake for all other neighbouring tags. For this reason, performance was very poor compared to the centralized approach. Hence, we revise this algorithm.

The revised localized algorithm is similar to the existing one, but makes scheduling decisions differently. The transition rules are as follows:

4.2.2 Transition rules

(a) Each reader starts in the awake state and writes its energy into neighbouring tags.
(b) Then, it checks the state of all neighbouring tags. It keeps two counts, one for readers with more energy than itself (sleep count), and one for readers with less energy (awake count). Any time it encounters another reader's energy value in a tag, it increases either of these counts by comparing its energy to the energy of the other reader.
(c) Once it has evaluated all neighbouring tags, it compares its two counts. If the sleep count is higher, it goes to sleep. Otherwise, it stays awake.
(d) Readers that are asleep have a random chance of waking up from sleep (p value). If they do, they go back into the checking state.
(e) Readers that are awake also have a random chance of going back to the checking state.
(f) This repeats until all readers have been exhausted of energy.

This approach results in more stable scheduling as it considers all neighbouring tags rather than sleeping as soon as it sees a tag better covered by another reader. Network lifetime is much higher than the first approach. This approach actually obtains very similar network lifetime to the centralized algorithm, and in some cases has even better tag coverage. Results are presented in the next section.

4.2.3 Example

Each reader begins with the same amount of energy. Each writes its energy into each neighbouring tag around that reader. It then goes into a checking state to see if the tag is covered by another readers, and how much energy they have. Since all readers start with the same amount of energy, the reader's ID is used as a tiebreaker. Each reader would keep count of how many readers had more energy than it (called the sleep count), and how many readers had less energy than it (called the awake count). Reader 5 would have an awake count of 4, since it has the highest ID. The rest of the readers would have a awake count of 0 and an sleep count of 1. Since the sleep conut is greater than the awake count, they would therefore all go to sleep until they randomly decided to wake up again. At that point they would follow the same process and obtain a count. At that point, however, they would likely have more energy than reader 5. Their awake count would then be 1 and sleep count would be 0, so they would switch to the awake state. Reader 5, at the same time, could have also randomly went back into the checking state and seen that the other readers had more energy, resulting in its sleep count being 4 and awake count being 0, and going to sleep. This process would repeat until all readers were exhausted of energy.

4.3 Localized algorithm 2

Another approach was implemented and tested, where readers do not write any information into the tag at all. Instead, they directly communicate with readers around them to determine if any are awake. As soon as they see a neighbouring reader that is awake, they go to sleep. Since readers need to send additional signals to readers in addition to interrogation signals to the tags, additional energy needs to be used. With this approach, however, the readers that remained awake would do so until they were exhausted of energy. This creates spikes in the graph where it clearly shows the readers all running out of energy at the same time and new ones switching to the awake state.

4.3.1 Transition rules

(a) Each reader randomly starts off either asleep or awake.
(b) While asleep, there is a random chance based on the p value that a reader will wake up and go back into checking.
(c) In the checking state, the reader simply checks if there are any readers awake in its immediate neighbourhood. If so, the reader goes back into the sleep state. If no other readers are awake, the reader will switch to the awake state.

4.3.2 Example

This localized algorithm is much simpler than the previous algorithms. Each reader simply checks around itself to see if there are any other readers awake using direct reader to reader communication. All readers would start awake and see that there were awake readers nearby, so would initially all go to sleep. While asleep, however, readers have a chance of waking up from sleep and going back into the checking state. By chance, reader 5 switches into checking, sees all neighbouring readers asleep, and switches to the awake state. It would remain on until it ran out of energy. Any other readers that randomly woke up during this time would go into checking and see reader 5 still awake, so would stay asleep. Once reader 5 ran out of energy, the next reader to wake up would see that there are no awake readers, and wake up. It would then run until it ran out of energy, and this would repeat until all readers ran out of energy.

4.4 Localized algorithm 3

The tendency for a reader to remain awake until it was completely exhausted was a frequent occurrence in the second localized algorithm. To prevent this behavior, a value was added to each reader to cause awake readers to eventually switch to the sleep state to allow other readers to switch to the awake state. This causes energy to be used more evenly among readers.

4.4.1 Transition rules

(a) All readers start awake.
(b) While awake, a number is randomly generated and compared to the current p value of the reader.
(c) If the value is less than the p value, the reader switches to the sleep state.
(d) Otherwise, the p value is incremented to increase future probability of switching to the sleep state.
(e) In the checking state, a reader checks its immediate neighbours to see if there are any awake readers. If so, it goes to sleep. If not, it stays awake.
(f) This repeats until all readers have run out of energy.

4.4.2 Example

All readers start awake. Each sees that all other readers are awake and all switch into the sleep state. While in the sleep state, there is a chance p that it will wake from sleep and go into checking. Perhaps reader 3 wakes up, sees that all neighbouring readers are asleep, and switches awake. However, reader 3 would not remain until it is exhausted of energy. On each time iteration, it generates a random value. If this value is less than p, the reader would switch to the sleep state. Otherwise, the p value is increased, meaning that it is more likely to go to sleep the longer it stays awake. This ensures that readers spend energy more evenly compared to the previous approach.

5 Performance evaluation

All results are obtained using a custom simulator written in Java. The simulator is based on a cellular automata model. On each time iteration, the simulator iterates through all readers and runs their scheduling method. Each reader makes a scheduling decision based on the algorithm and sets its state and deducts the specified amount of energy based on what state it is in. The number of tags being covered is output on each iteration. Simulation ends when all readers have run out of energy.

A topology generator was created that randomly places readers and tags in a specified size grid. It also allows the ratio of readers and tags to be modified. By generating a variety of random scenarios, performance of algorithms in various scenarios can be determined.

Each set of simulations uses a different proportion of readers and tags on a 100×100 grid. The first has an equal number of readers to tags, the second has 25% readers, and the last has 15% readers. While the centralized algorithm usually finds a better solution, this is not always the case as it is only an approximation to an NP-hard problem. The localized algorithm, however, still obtains stable tag coverage throughout network lifetime, and sometimes achieves better tag coverage than the centralized algorithm. The second and third localized algorithms achieve shorter network lifetimes because of the additional energy required for the direct reader communication.

Additionally, area under the curve calculations can also be used to compare algorithm performance.

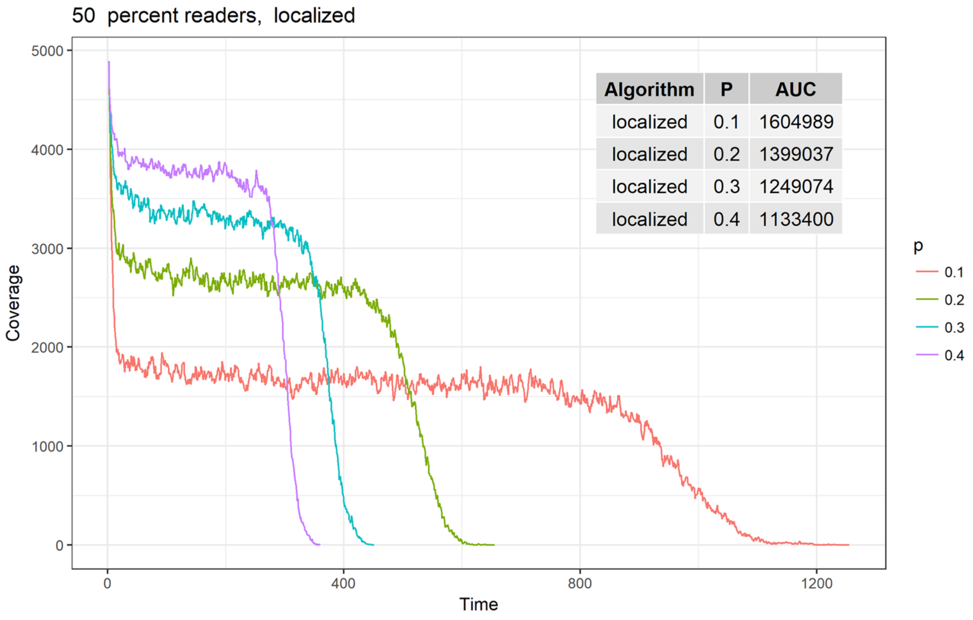
5.1 Simulation environment

Simulations were run on both a ThinkPad T440s with a Core i5 processor and 8GB of RAM, and a desktop with a Core i7 4770k with 8GB of RAM.

5.2 Discussions and results

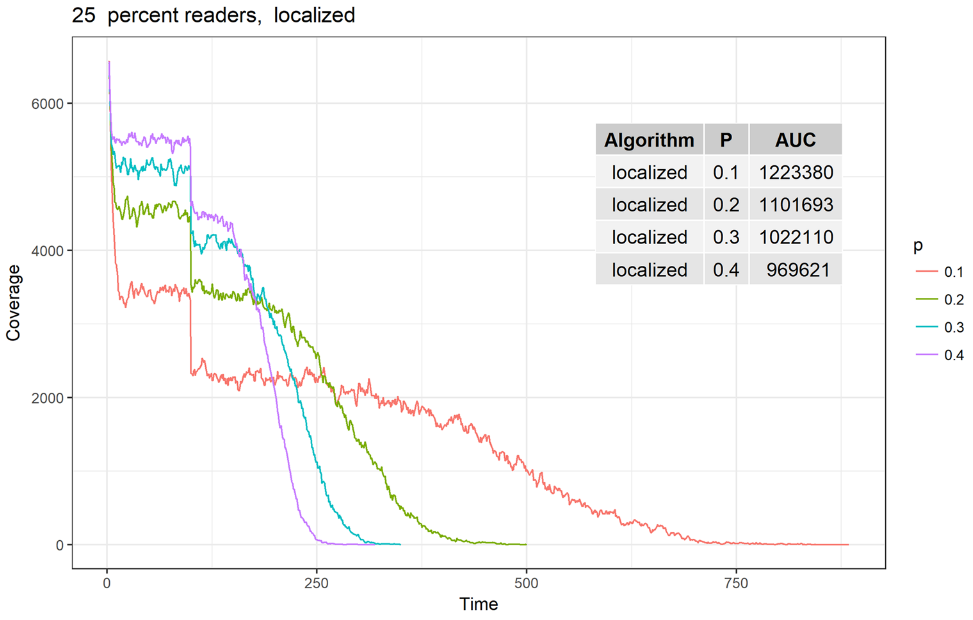
The following Figs. (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13) show the performance of all three localized algorithms for various scenarios. For all scenarios, p value 0.1 lasts the longest since the readers are not as likely to switch between states, which uses additional energy.

Each graph in this section shows interesting behaviours of each of the algorithms presented in this paper. Although the centralized algorithm usually produces an optimal scheduling, there are several limitations to it. Since it only runs once at the beginning of the simulation, new readers cannot be scheduled as they join the network. Or, if a reader is mobile and moves locations, the schedule generated by



Algorithm	P	AUC
localized	0.1	1604989
localized	0.2	1399037
localized	0.3	1249074
localized	0.4	1133400

Fig. 2 Localized algorithm, 100 × 100 grid, 50% readers



Algorithm	P	AUC
localized	0.1	1223380
localized	0.2	1101693
localized	0.3	1022110
localized	0.4	969621

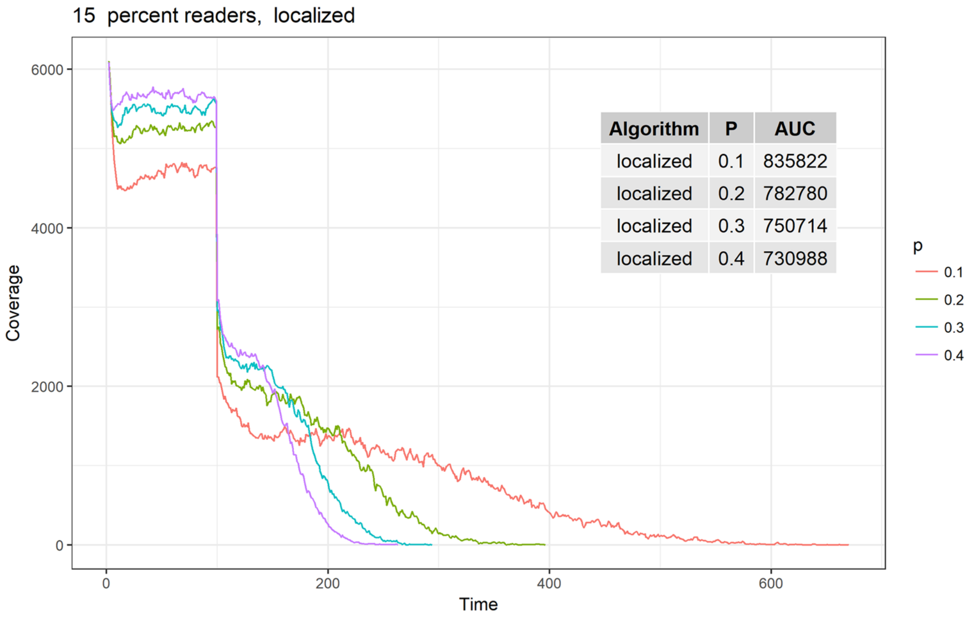
Fig. 3 Localized algorithm, 100 × 100 grid, 25% readers

the algorithm may no longer be optimal. With the localized algorithms, scheduling occurs throughout the simulation. It can therefore accept readers being added, moved, or removed. In addition, the localized algorithms are simpler to set up as no central node is required to make scheduling decisions. Each reader, once activated, communicates with other readers via tag memory to make scheduling decisions.

It is also interesting to note the effect of the p value on the localized algorithms. When p values are lower, the readers are less likely to wake from sleep or go into checking while they are awake. Tag coverage is slightly lower at first, but network lifetime is longer. In a situation where tag coverage is more important than network lifetime, the p value can be increased to meet requirements. The centralized algorithm does not require a p value, as there is no randomness to it; it repeatedly places the reader with the highest tag coverage into the current disjoint set. It will always produce the same output given a particular input.

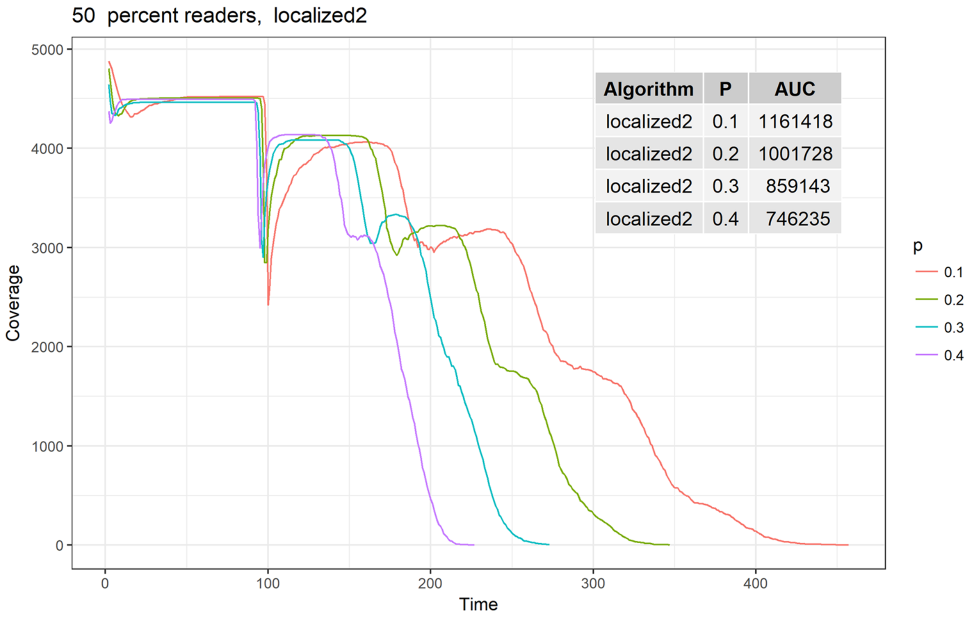
By observing the area under the curve calculations for each localized algorithm, it is clear that a p value of 0.1 provides the best overall performance. In the comparisons between localized and centralized algorithms, the localized algorithm achieved nearly the same performance as the centralized in the scenario with 50% readers (Fig. 11), and better performance in scenarios with 25 (Fig. 12) and 15% (Fig. 13) readers.

Fig. 4 Localized algorithm, 100 × 100 grid, 15% readers



Algorithm	P	AUC
localized	0.1	835822
localized	0.2	782780
localized	0.3	750714
localized	0.4	730988

Fig. 5 Localized algorithm 2, 100 × 100 grid, 50% readers

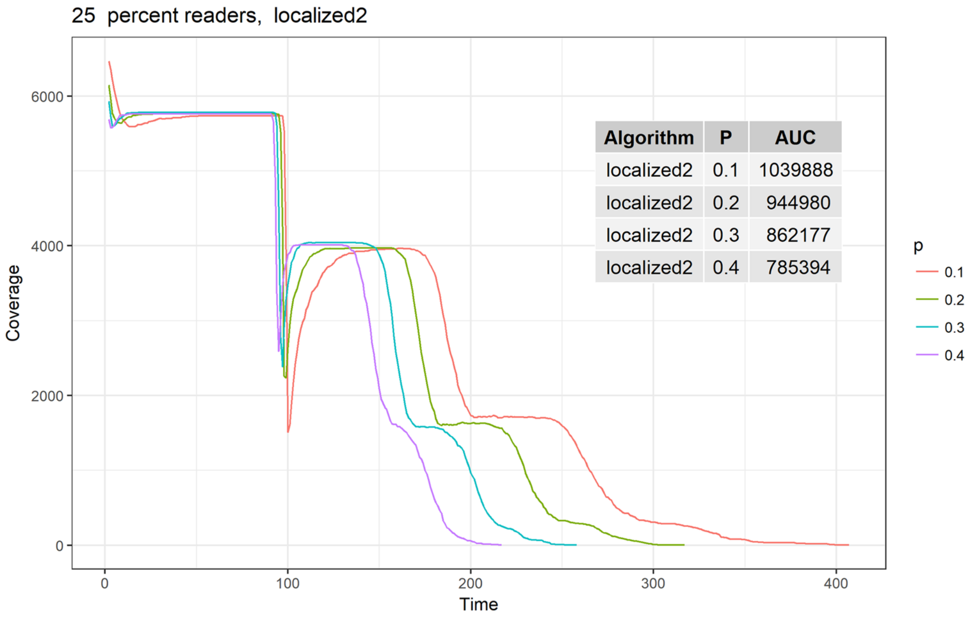


Algorithm	P	AUC
localized2	0.1	1161418
localized2	0.2	1001728
localized2	0.3	859143
localized2	0.4	746235

Another observation noticed is that the fewer readers there are, the closer the performance is between centralized and localized algorithms. In scenarios with larger numbers of readers, the centralized algorithm has more flexibility when choosing readers for each disjoint set. However, when there are few readers, there are fewer possible disjoint sets. Each reader has to cover a great deal of tags, so they are quickly exhausted of energy. With the localized algorithms, readers can be scheduled more flexibly to provide more stable tag coverage over a period of time. This is why the area under the curve readings are so close between centralized and localized in the graphs with 25% and 15% readers.

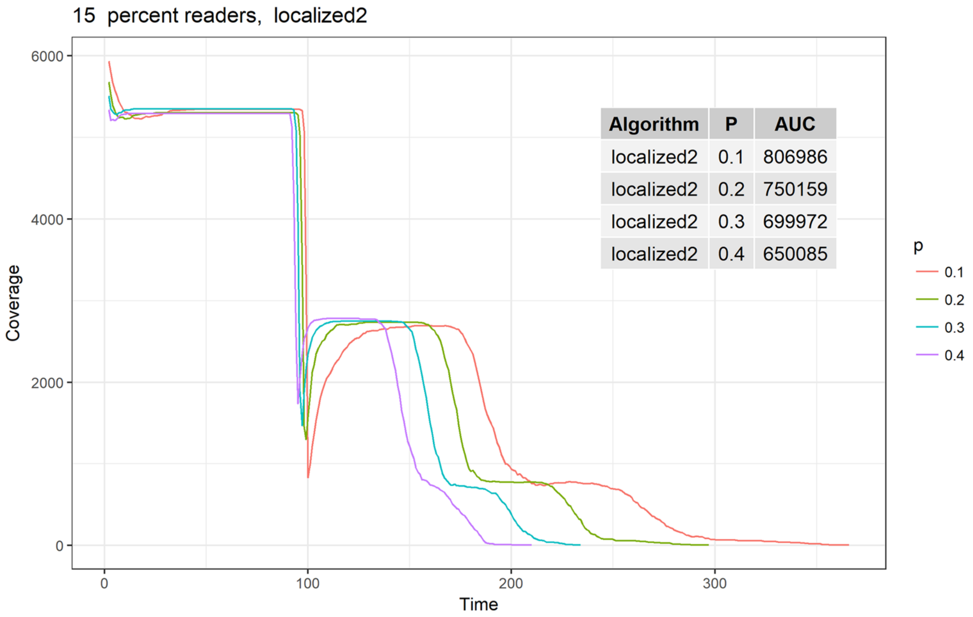
In addition to increasing network lifetime, these algorithms also help reduce collisions. Readers use the transition rules in the algorithm to go to sleep if they detect that they do not need to be awake in the current time iteration. This prevents two readers covering the same tag to transmit at the same time.

Using these algorithms, lifetime of mobile RFID readers can be increased. This reduces the rate at which batteries discharge, resulting in less downtime due to charging, or batteries needing to be replaced, resulting in reduced costs and waste. This will make RFID a practical choice for future tracking applications where readers need to be mobile.



Algorithm	P	AUC
localized2	0.1	1039888
localized2	0.2	944980
localized2	0.3	862177
localized2	0.4	785394

Fig. 6 Localized algorithm 2, 100 × 100 grid, 25% readers



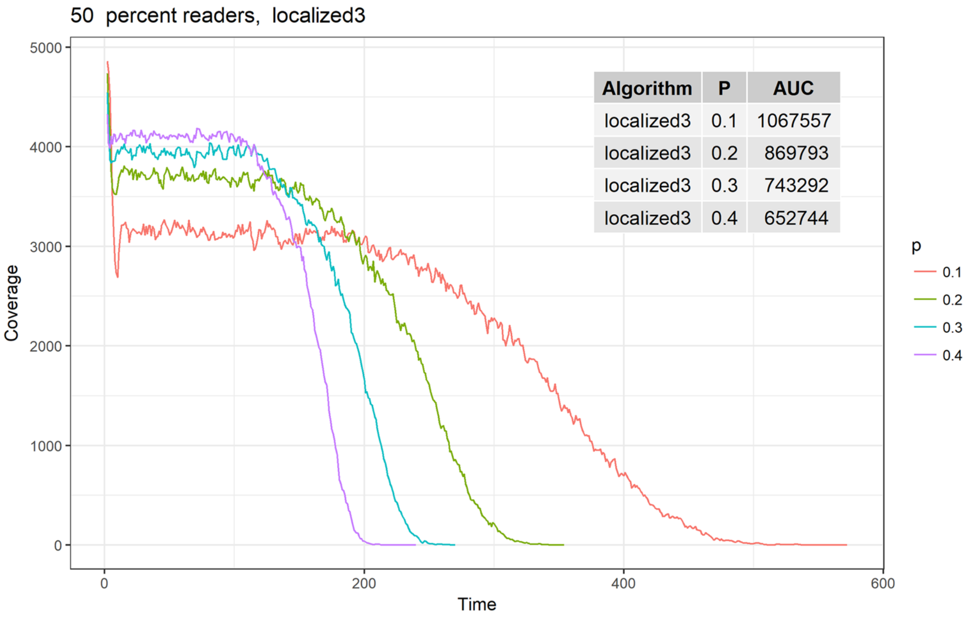
Algorithm	P	AUC
localized2	0.1	806986
localized2	0.2	750159
localized2	0.3	699972
localized2	0.4	650085

Fig. 7 Localized algorithm 2, 100 × 100 grid, 15% readers

6 Conclusion

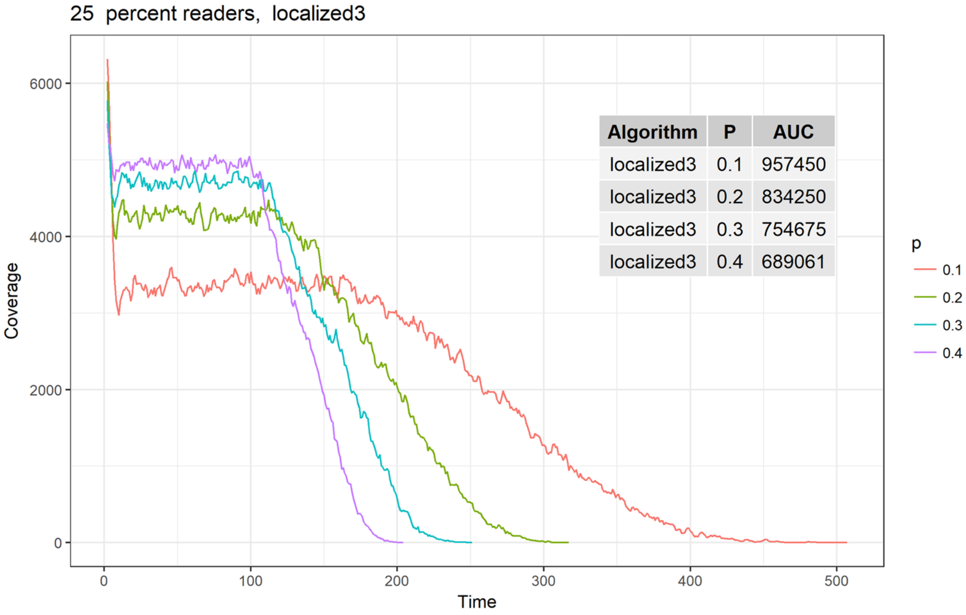
We considered scheduling of readers in RFID networks where the networking lifetime of the network and the tag coverage are maximized. We designed various localized Using the algorithms to solve the problem. Extensive simulation results show that a localized RFID reader scheduling algorithm can achieve comparable or better performance than a centralized algorithm in terms of network lifetime. After reviewing techniques found in various existing scheduling algorithms in RFID, a simulator was developed to implement and test new energy scheduling algorithms. Using that simulator, a variety of simulations were run on scenarios with different characteristics. The resulting graphs and calculations show that the proposed localized algorithm achieve similar or better results in a variety of scenarios. In our future work, we aim at implementing these algorithms on physical RFID readers and test their performance in reality. In addition, considering mobile tags and readers in designing such algorithms are interesting research topics.

Fig. 8 Localized algorithm 3, 100 × 100 grid, 50% readers



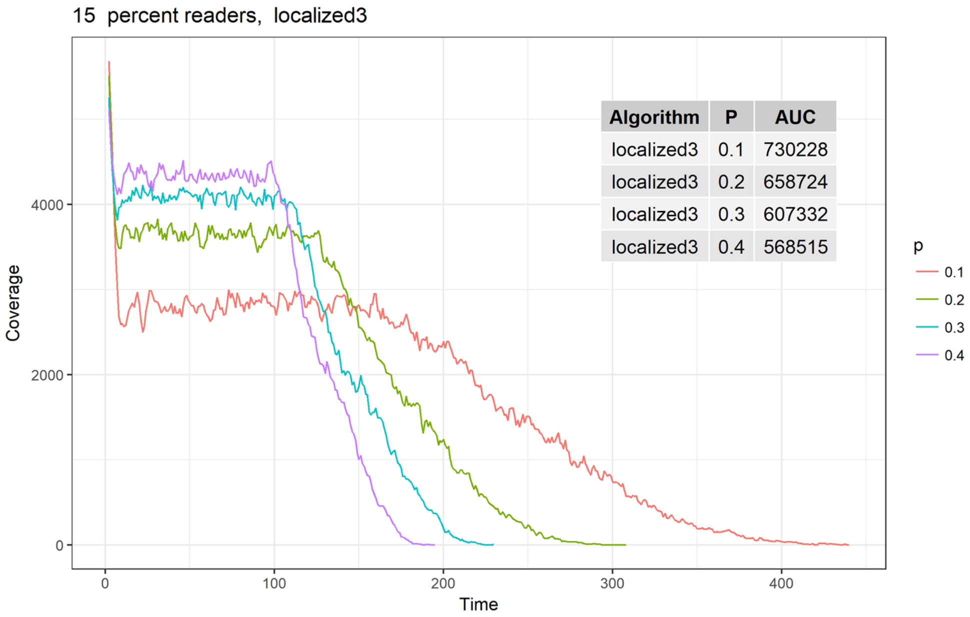
Algorithm	P	AUC
localized3	0.1	1067557
localized3	0.2	869793
localized3	0.3	743292
localized3	0.4	652744

Fig. 9 Localized algorithm 3, 100 × 100 grid, 25% readers



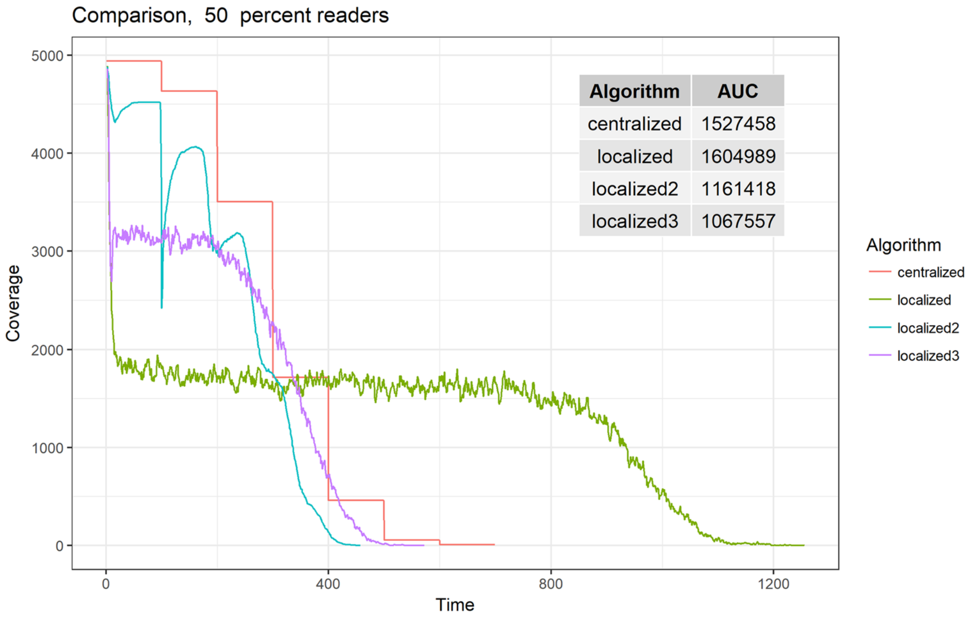
Algorithm	P	AUC
localized3	0.1	957450
localized3	0.2	834250
localized3	0.3	754675
localized3	0.4	689061

Fig. 10 Localized algorithm 3, 100 × 100 grid, 15% readers

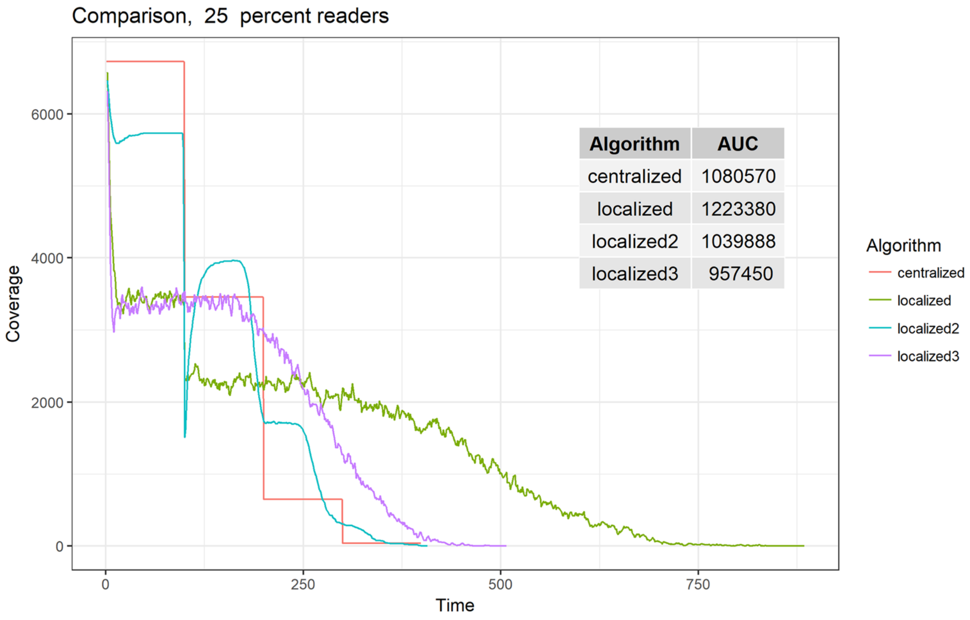


Algorithm	P	AUC
localized3	0.1	730228
localized3	0.2	658724
localized3	0.3	607332
localized3	0.4	568515

Fig. 11 Centralized vs Localized algorithms (P: 0.1), 50% readers

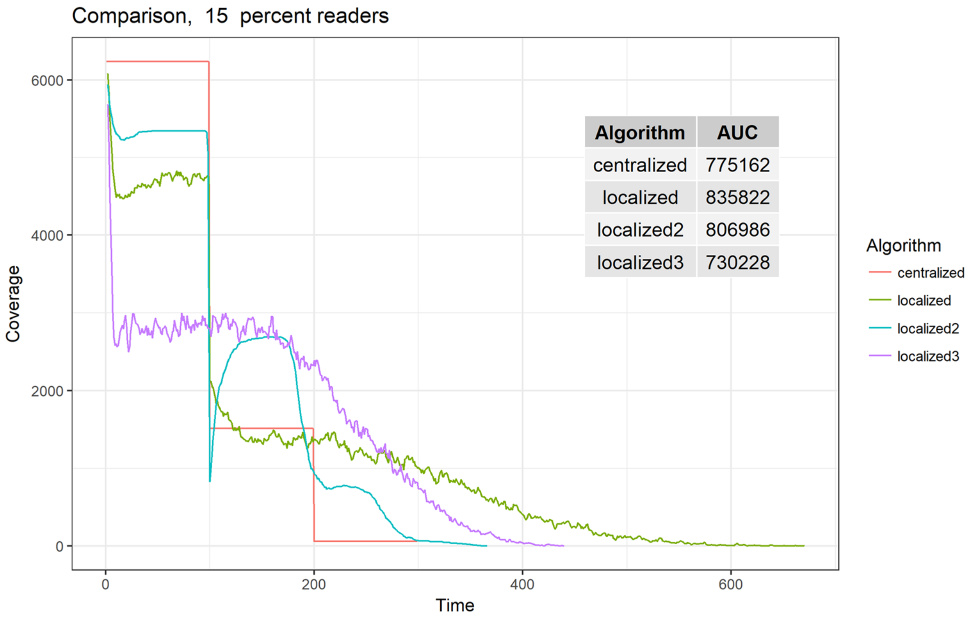


Algorithm	AUC
centralized	1527458
localized	1604989
localized2	1161418
localized3	1067557



Algorithm	AUC
centralized	1080570
localized	1223380
localized2	1039888
localized3	957450

Fig. 12 Centralized vs Localized algorithms (P: 0.1), 25% readers



Algorithm	AUC
centralized	775162
localized	835822
localized2	806986
localized3	730228

Fig. 13 Centralized vs Localized algorithms (P: 0.1), 15% readers

References

Bueno-Delgado MV, Pavón-Mariño P (2013) A maximum likelihood-based distributed protocol for passive RFID dense reader environments. J Supercomput 64(2):456–476. https://doi.org/10.1007/s11227-012-0779-5

Cardei M, Du DZ (2005) Improving wireless sensor network lifetime through power aware organization. Wireless Netw 11(3):333–340. https://doi.org/10.1007/s11276-005-6615-6

Chen NK, Chen JL, Lee CC (2009) Array-based reader anti-collision scheme for highly efficient RFID network applications. Wireless Commun Mobile Comput 9(7):976–987. https://doi.org/10.1002/wcm.646

Choudhury S (2012) Cellular automaton based algorithms for wireless sensor networks. Queen's University, Kingston

Choudhury S, Akl SG, Salomaa K (2012a) Energy efficient cellular automaton based algorithms for mobile wireless sensor networks. In: 2012 IEEE wireless communications and networking conference (WCNC), Shanghai, pp 2341–2346. https://doi.org/10.1109/WCNC.2012.6214185

Choudhury S, Salomaa K, Akl SG (2012b) A cellular automaton model for connectivity preserving deployment of mobile wireless sensors. In: 2012 IEEE international conference on communications (ICC), Ottawa, ON, pp 6545–6549. https://doi.org/10.1109/ICC.2012.6364914

Choudhury S, Salomaa K, Akl SG (2012c) A cellular automaton model for wireless sensor networks. J Cell Automata 7(3):223–241

Choudhury S, Salomaa K, Akl SG (2014) Cellular automaton-based algorithms for the dispersion of mobile wireless sensor networks. Int J Parallel Emerg Distrib Syst 29(2):147–177

Chvatal V (1979) A greedy heuristic for the set-covering problem. Math Oper Res 4(3):233–235. https://doi.org/10.1287/moor.4.3.233.

Domdouzis K, Kumar B, Anumba C (2007) Radio-Frequency Identification (RFID) applications: a brief introduction. Adv Eng Inform 21(4):350–355. https://doi.org/10.1016/j.aei.2006.09.001

El Fissaoui M, Beni-Hssane A, Saadi M (2018) Energy efficient and fault tolerant distributed algorithm for data aggregation in wireless sensor networks. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-018-0704-8

Garzon M (2012) Models of massive parallelism: analysis of cellular automata and neural networks. Texts in theoretical computer science. An EATCS Series. Springer, Berlin. https://books.google.ca/books?id=e8OqCAAAQBAJ. Accesed Feb 2018

Hassan MY, Hussain F, Choudhury S (2018) Connectivity preserving obstacle avoidance localized motion planning algorithms for mobile wireless sensor networks. Peer-to-Peer Netw Appl. https://doi.org/10.1007/s12083-018-0656-y

Kumar V, Kumar A (2018a) Improved network lifetime and avoidance of uneven energy consumption using load factor. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-018-0857-5

Kumar V, Kumar A (2018b) Improving reporting delay and lifetime of a WSN using controlled mobile sinks. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-018-0901-5

Lounis M, Bounceur A, Laga A, Pottier B (2015) GPU-based parallel computing of energy consumption in wireless sensor networks. 2015 Eur Confer Netwo Commun EuCNC). https://doi.org/10.1109/EuCNC.2015.7194086. http://ieeexplore.ieee.org/document/7194086/. Accesed Feb 2018

Lozano-Nieto A (2013) RFID DESIGN fundamentals and applications, vol 53. CRC press, Boca Raton, Florida. https://doi.org/10.1017/CBO9781107415324.004. arXiv:1011.1669v3

Natarajan K, Prasath B, Kokila P (2016) Smart health care system using internet of things. J Network Commun Emerg Technol (JNCET) 6(3):37–42. https://doi.org/10.15680/IJIRCCE.2016

Rashid N, Choudhury S, Salomaa K (2016) CARRE: cellular automaton based redundant readers elimination in RFID networks. In: 2016 IEEE international conference on communications (ICC), Kuala Lumpur, pp 1–6. https://doi.org/10.1109/ICC.2016.7510604

Rashid N, Choudhury S, Salomaa K (2018) Localized algorithms for redundant readers elimination in RFID networks. Int J Parallel, Emergent Distrib Syst. https://doi.org/10.1080/17445760.2017.1419242

Vales-Alonso J, Parrado-García FJ, Alcaraz JJ (2016) OSL: an optimization-based scheduler for RFID Dense-Reader Environments. Ad Hoc Netw 37:512–525. https://doi.org/10.1016/j.adhoc.2015.10.004

Waldrop J, Engels DW, Sarma SE (2003) Colorwave: a MAC for RFID reader networks. IEEE Wireless Commun Netw Confer WCNC 3:1701–1704. https://doi.org/10.1109/WCNC.2003.1200643

Wang YC, Liu SJ (2017) Minimum-cost deployment of adjustable readers to provide complete coverage of tags in RFID systems. J Syst Softw 134:228–241. https://doi.org/10.1016/j.jss.2017.09.015

Yang J, Liu F, Cao J (2017) Greedy discrete particle swarm optimization based routing protocol for cluster-based wireless sensor networks. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-017-0515-3

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.