Location Information Dissemination Scheme for RFID-based Distributed Localization Systems

Lobna M. Eslim, Hossam S. Hassanein and Walid M. Ibrahim
Telecommunications Research Lab
School of Computing, Queen's University, Kingston, ON, Canada K7L 3N6
{lobna.eslim, hossam, walid}@cs.queensu.ca

Abstract—**The availability of location information is essential for context- and location-aware services, which are typically provided by a large number of applications. RFID systems are extensively utilized to provide localization service typically through a centralized and coordinated approach. In this paper, we propose a distributed location information dissemination scheme using heterogeneous uncoordinated mobile RFID readers with the support of inexpensive "memory spots". In the proposed scheme, mobile RFID readers localize passive RFID-tagged objects and leverage the available memory spots in a given smart environment to disseminate location information. Mobile RFID readers use such memory spots to store tag locations and queries enabling exchange location information without the need for direct communication among each other. We study the behavior of the proposed scheme and compare its performance with a typical pull dissemination strategy through extensive simulations using ns-3. Our results indicate that the proposed scheme outperforms the typical pull dissemination strategy in terms of localization delay and average overhead under different dynamicity settings.**

I. INTRODUCTION

The Internet of Things (IoT) can be broadly defined as a decentralized system of smart objects, which are seamlessly integrated into an information network to provide a variety of smart services over the Internet [1] [2]. This vision unlocks the door for a new generation of applications and solutions which span a wide and diverse range of domains (e.g. smart environments, healthcare, transportation, manufacturing, etc.) [3]. IoT applications are inherently distributed as they are built on spatially disseminated smart objects to offer users more convenient context- and location-aware services. Locating smart objects along with the availability of the location information is a key requirement for IoT applications. In the literature, many wireless technologies such as Infrared, Ultrasound, Radio Frequency Identification (RFID), Ultra Wide Band (UWB) and Bluetooth [4] have been leveraged to provide localization service. Recently RFID, which is one of the key enabling technologies in IoT [5], has attracted significant research interests. RFID has a number of unique features rendering it a strong candidate to provide localization service in an inexpensive, reliable, flexible and scalable manner – key requirements in IoT applications.

A typical RFID system is composed of a set of tags (passive, semi-passive or active) and a set of readers [6]. Several RFID localization systems have been proposed for mobile/stationary tagged objects or mobile readers [7]. For tagged-object localization, objects are localized through a set of coordinated readers that report to a central server for location estimation. While such systems are cost effective, they are based on a centralized and fixed infrastructure which provides limited scalability and may not be plausible in typical IoT large-scale dynamic scenarios.

In this paper, we propose a distributed RFID-based localization and location dissemination scheme to localize mobile or stationary passive RFID tagged-objects using heterogeneous uncoordinated mobile RFID readers. The system uses a minimal inexpensive and flexible component known as "memory spots". A memory spot is limited storage and processing device that can use WiFi, Bluetooth, Zigbee or RFID technology. Memory spots are used to disseminate passive RFID tagged-objects' locations and to exchange location queries; providing high location information availability with lower overhead[1]. In the proposed scheme, readers periodically interrogate tags in their vicinities and synchronize tags' location information with memory spots they may pass by. In the synchronization process, a reader: (1) updates the memory spot with interrogated tags, (2) obtains location of tags beyond its interrogation zone and (3) either replies to or carries on and propagates location queries that may exist. Carrying a query allows rapid propagation of such query towards other memory spots. Readers interested in the location of a tag can interrogate the nearest memory spot to pull location information obtained from other passing readers, or to register a location query. Specifically, we contribute by:

- leveraging heterogeneous uncoordinated mobile RFID readers to localize surrounding tags through indirect cooperation;
- introducing the concept of "memory spots" – a minimal inexpensive and flexible component to maintain location information availability; and
- designing a distributed protocol for location information dissemination amongst RFID readers through the memory spots.

The remainder of this paper is organized as follows: Section II reviews the related work and motivates our approach. Section III defines our assumptions and explains in detail our proposed scheme. Section IV discusses some practical considerations. Section V presents performance evaluation and results analysis. Finally, conclusions are provided in Section VI.

---

[1] For convenience, we use the terms tag and reader to represent passive RFID-tagged object and mobile RFID reader, respectively.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

II. RELATED WORK

RFID systems have been extensively utilized to localize mobile or stationary objects, these are broadly known as tag localization systems. In addition, the possibility of having shared memories in a given environment persuaded researchers to take advantage of such resources in data exchange [8], smart transportation [9] and cooperative localization [10].

In tag localization systems, objects are identified by RFID tags either active or passive to be localized through a set of coordinated RFID readers which detect tags and report detection information to a central server for location estimation [11]-[16]. Some systems support coordinated readers by a typically large set of reference tags to enhance the location estimation as in LANDMARC [11] and its enhancement in [12]. In these systems, the location server compares a tag's Received Signal Strength (RSS) at the readers with those of reference tags and estimates the tag location based on the locations of the k-nearest reference tags. Similarly, the schemes VIRE [14] and L-VIRT [15] use virtual reference tags instead of dense deployment of reference tags. VIRE [14] for instance, calculates the RSS of each virtual reference tag based on the RSS of surrounding reference tags. Then, the scheme compares a tag's RSS to those of reference tags either real or virtual, obtains all credible locations and filters them using an elimination algorithm. The use of mobile readers to localize tagged objects is proposed in [13] and [16] with support of landmarks, where reader-tag distance and tag-landmark distance are used to estimate a tag's location. Such systems, especially those based on passive RFID components, are cost effective. However, central- and fixed infrastructure-based solutions provide limited scalability and may not be practical in dynamic and/or mobile IoT settings. To seek scalability, the study in [17] leverages a set of heterogeneous mobile RFID readers to cooperatively localize tagged-objects in a distributed manner, and introduces a protocol for timely dissemination of location information among the mobile RFID readers.

Recently, shared resources by means of residual memories on passive RFID tags have been proposed for different purposes. For instance, reference [8] proposes a new paradigm to exchange data among a group of RFID readers by using tags' resources as a virtual channel. Similarly, the approach in [10] uses tags' resources to store spatial information obtained from different readers for distributed localization. The objective of our work is to design a distributed location information dissemination scheme for RFID-based distributed localization systems; working on increasing location information availability with less overhead using minimal inexpensive infrastructure.

III. LOCATION INFORMATION DISSEMINATION SCHEME

Our approach aims to provide high location information availability in RFID-based distributed localization systems with the support of a minimal inexpensive and flexible infrastructure. Our scheme relies on memory spots that are typically distributed in smart areas to be used by ad hoc readers that are mobile and uncoordinated to disseminate tags' locations and exchange location queries. Fig. 1 shows the general framework of our proposed scheme. The following subsections detail the scheme's components, assumptions along with the exchanged information and scheme's operation.

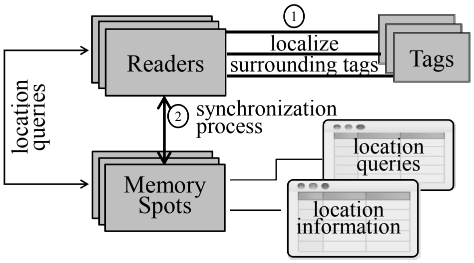


Fig. 1: General framework of the proposed scheme.

A. Scheme Components and Assumptions

As represented in Fig. 1, the operation of our proposed scheme depends on three main entities: (1) *Tags*, representing the objects to be localized, which can be either stationary or mobile and are identified by passive RFID tags, (2). *Memory Spots*, which represent the available memory spots in a given environment offering information storage and retrieval and (3) *Readers*, representing the ad hoc uncoordinated, dynamic and heterogeneous RFID readers, which can localize surrounding *Tags* and update *Memory Spots* accordingly. Each *Memory Spot* holds two types of information: location information and location queries. Location information contains the estimated positions of interrogated *Tags*. This information is created and periodically updated by passing *Readers*. Location queries hold queries about *Tags* that are of interest and need to be localized. Location queries information is updated either by a *Reader* communicating with a *Memory Spot* to pull location information, which is not currently available or by a passing *Reader* carrying a location query from another *Memory Spot*. In this work, we assume that *Readers*, as mobile devices, can acquire their own locations via one of the positioning systems for mobile devices. Readers can interrogate surrounding tags and communicate with and update all shared *Memory Spots*.

B. Scheme Operation

We define two events: *localize* and *synchronize* which are conducted by *Readers* such that each *synchronize* event may witness multiple *localize* events. At each *localize* event, a *Reader* interrogates surrounding tags; generating time-stamped location information about tags in its vicinity. At each *synchronize* event, a *Reader* communicates with any *Memory Spot* it may pass by, updates the location information on such *Memory Spot* and carries location information that needs to be disseminated for interests of other *Readers*. In addition to the two aforementioned events, we define an occasional event, named *query*, which is conducted according to a need. When a *Reader* is interested in localizing a *Tag(s)* beyond its interrogation zone, it checks any *Memory Spot* it may pass by to pull the required location information or to submit a location query to be manipulated during coming *synchronize* events. Fig. 2 explains the schema of location information and location queries on both *Readers* and *Memory Spots*. We describe the *Tags* localization, location query, and synchronization process from the perspective of both the *Reader* and the *Memory Spot* along with an illustrative example.

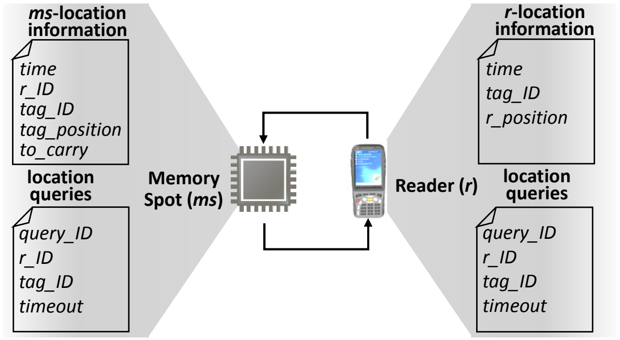


Fig. 2: Schema of location information and location queries on both *Memory Spots* and *Readers*.

1) Tags localization

In this process, *Readers* maintain location information about *Tags* in their interrogation zones through a sequence of *localize* events. This information may range from simple proximity location to more accurate location based on *Readers* capabilities (i.e. adaptive power, antenna array, etc. [18]). For simplicity, we consider only proximity location information which contains: *time, tag ID, and a reader position*. This produces location information on each and every *Reader* which needs to be aggregated, disseminated and be available in a timely fashion.

2) Location query

This process is to run by a *Reader* when it is interested in localizing a tag(s) out of its current interrogation zone. As explained in Algorithm I, a *Reader* interested in a tag's location, first looks for the tag of interest in its local location information. If the required location information does not exist, the *Reader* starts communicating with *Memory Spots* it passes by to either pulling the required location information or registering a location query. When registering a location query, the *Reader* generates a random ID (*query_ID*) in addition to its ID (*r_ID*) to identify its query, and assigns a *timeout* after which it may not be interested in localizing such tag(s).

Algorithm I Location query Algorithm - run by *Readers*

Input: tag ID **Output:** tag's location information

```
set loc(Reader_ID) = Reader_ID.location information
set loc(Memory Spot_ID) = Memory Spot_ID.location information
set query(Memory Spot_ID) = Memory Spot_ID.location queries
if tag_ID exists in loc(Reader_ID) then
    set not localized = False
    return tag_ID.position
else
    set not_localized = True
    while not_localized do
        contact Memory Spots
        for each contacted Memory Spot do
            if tag_ID exists in loc(Memory Spot_ID) then
                set not localized = False
                return tag_ID.position
            else
                generate Q(query_ID, Reader_ID, Tag_ID, timeout)
                add Q to query(Memory Spot_ID)
            end if
        end for
end if
```

3) Synchronization process

The main process in our proposed location information dissemination scheme is to iteratively synchronize location information on the distributed *Memory Spots* by passing *Readers*. When a *Reader* conducts a *synchronize* event and accordingly communicates with a *Memory Spot*, the following actions take place:

1. The *Reader* updates the *Memory Spot*'s location information and acquires locations of tags it may be interested in.
2. The *Reader* pushes its location queries into the *Memory Spot*, which are carried from other *Memory Spots* through previous *synchronize* events.
3. The *Memory Spot* filters its location queries and discards any query that has been answered or has expired. Accordingly, the *Memory Spot* switches the flag named "*to_carry*" in its location information (Fig. 2) for tags in answered queries to be disseminated by passing *Readers*.
4. The *Reader* carries both "*to_carry*" location information and remaining queries from the *Memory Spot* to propagate them into other *Memory Spots* that it may pass by.

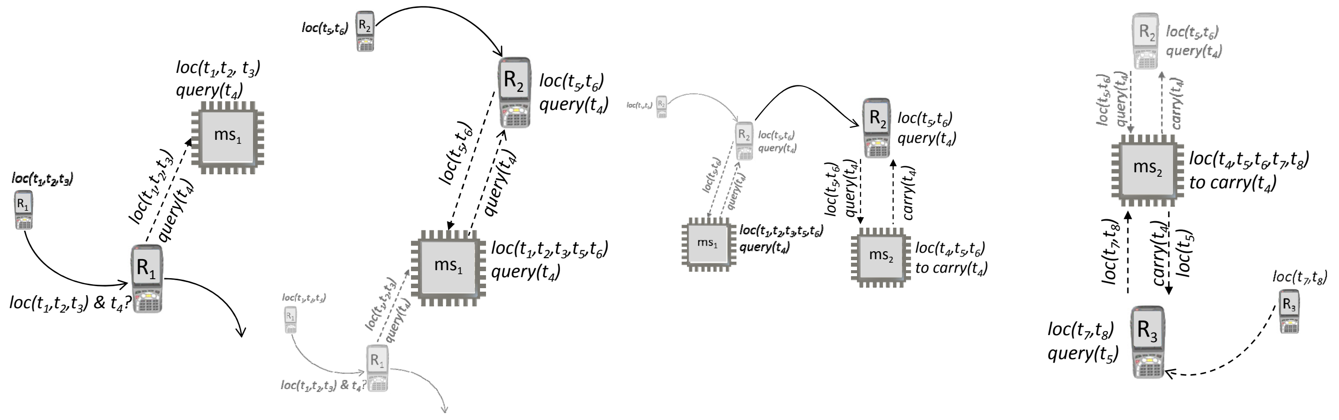
Algorithm II details how location information is updated by passing *Reader* on the contacted *Memory Spot* (i.e. action 1), and it shows how the *Reader* pushes the queries that it holds into the *Memory Spot* to be answered by any other passing *Reader* (i.e. action 2). During the update, the most recent location information is considered in the event of conflict. However, in case of having more accurate location estimation technique than proximity; the accuracy has to be considered as well to decide which location information to use.

Algorithm II Loc/query information updating Algorithm - run by *Readers*

Input: loc/query information **Output:** updated loc/query information

```
set loc(Reader_ID) = Reader_ID.location information
set query(Reader_ID) = Reader_ID.location queries
set loc(Memory Spot_ID) = Memory Spot_ID.location information
set query(Memory Spot_ID) = Memory Spot_ID.location queries
for each record $r_i$ in loc(Reader_ID) do
    if ($r_i$.tag_ID is not exist in loc(Memory Spot_ID) or $r_i$.time is most
        recent) then
            add ($r_i$.time, Reader_ID, $r_i$. tag_ID, $r_i$.r_position) to
                loc(Memory Spot_ID)
    end if
end for
for each query $q_i$ in query(Reader_ID) do
        add ($q_i$.query_ID, Reader_ID, $q_i$.tag_ID, $q_i$.timeout) to
                query(Memory Spot_ID)
end for
```

Running Algorithm II resulted in location queries information on the *Memory Spot* which contains answered, unanswered or expired queries that need to be manipulated. In addition, the *Memory Spot* is required to highlight the location information for the queries that have been answered to be carried and disseminated to serve interests of other *Readers;* regardless of the interests of current *Reader*. This takes place by switching the *to-carry* flags of these tags to the value of *1*. Algorithm III illustrates how the aforementioned steps generally run by *Memory Spots*, which represents the third action in the synchronization process.



(a) (b) (c) (d)

Fig. 3: Illustrative example. (a) R_1 updates ms_1 and register a query about t_4. (b) R_2 updates ms_1 and carry the query generated by R_1. (c) R_2 updates ms_2 and disseminates the query generated by R_1, accordingly ms_2 marks t_4 as "*to_carry*" thus, R_2 carries t_4. (d) R_3 updates ms_2, looks for t_5 and carries t_4 as well.

Algorithm III Queries filtration Algorithm - run by *Memory spot*

Input: location queries **Output:** filtered location queries

```
set loc(Memory Spot_ID) = Memory Spot_ID.location information
set query(Memory Spot_ID) = Memory Spot_ID.location queries
for each query $q_i$ in query(Memory Spot_ID) do
    if $q_j$.tag_ID is exist in loc(Memory Spot_ID) then
        set loc(Memory Spot_ID) → $q_j$.tag_ID.to_carry = 1
        delete $q_j$
    else if $q_j$.timeout < 0 then   delete $q_j$
    end if
end for
```

Running Algorithm III helps in releasing both the *Memory Spot* resources and the passing *Readers* communication overhead to serve only unanswered queries and stop propagating useless ones. Algorithm IV explains the last action in the synchronization process (i.e. action 4) to be run by passing *Readers* for the purpose of information dissemination. In this algorithm, a *Reader* carries both the remaining unanswered queries in addition to the "*to-carry*" location information to be considered in the next synchronization process. To avoid any loopback, the queries that are originally created by such *Reader* are ignored. When a *Reader* updates its location information, it only considers time-stamped tags locations irrespective of the reader that localized such tags.

Algorithm V Carry location queries Algorithm - run by *Reader*

Input: location queries **Output:** updated location queries

```
set loc(Reader_ID) = Reader_ID.location information
set query(Reader_ID) = Reader_ID.location queries
set loc(Memory Spot_ID) = Memory Spot_ID.location information
set query(Memory Spot_ID) = Memory Spot_ID.location queries
for each record $ms_j$ in loc(Memory Spot_ID)  do
    if $ms_j$.to_carry = 1 then
        add ($ms_j$.time, $ms_j$.tag_ID, $ms_j$.tag_position) to
            loc(Reader_ID)
    end if
end for
for each query $q_i$ in query(Memory Spot_ID)  do
    if ($q_i$.r_ID ≠ Reader_ID ) then
        add ($q_i$.query_ID, $q_i$.r_ID $q_i$.tag_ID, $q_i$.timeout) to
            query(Reader_ID)
    end if
end for
```

Fig. 3 shows an illustrative example for the operation of our proposed scheme. In Fig. 3 (a), the reader R_1 localized tags: t_1, t_2, t_3 and it was interested in localizing t_4. Then, R_1 communicated with the nearest memory spot ms_1, updated ms_1 with locations of t_1, t_2, t_3 and registered a query asking about t_4. The reader R_2 was in the vicinity of ms_1 as well (see Fig. 3 (b)). So, R_2 updated ms_1 with locations of t_5, t_6 and carried the query that was generated by R_1 about t_4. As in Fig. 3 (c), R_2, while it moves, communicated with memory spot ms_2 and did the following: updated ms_2 with locations of t_5, t_6, pushed the query about t_4 into ms_2. ms_2 had the location of t_4 which was previously updated by another passing reader, therefore ms_2 turned the *to_carry* flag of t_4 location to *1*. Accordingly, R_2 carried this location information to disseminate it across other memory spots. As in Fig. 3 (d), at ms_2, another reader R_3 was interested in locations that was updated by R_2, so it acquired such information (location of t_5) without query registration. In addition, R_3 updated ms_2 with locations of t_7, t_8 and carried the location of t_4 as well. R_1 while it moves, communicated with a memory spot that has been updated by either R_2 or R_3 hence, it acquired the location of t_4.

IV. PRACTICAL CONSIDERATIONS

Our approach suggests that in typical IoT settings, mobile RFID readers along with the available memory spots can be leveraged to provide localization service. This is achieved by allowing the mobile readers to localize surrounding tags[2], disseminate location information and exchange location queries indirectly by means of memory spots. From the practical perspective two points arise: the popularity of mobile RFID readers which can acquire their locations and the availability of memory spots in a given environment. The popularity of "mobile RFID readers" is a result of rapid adoption of embedded RFID readers in mobile devices due to the great interest by RFID manufacturers and the rapid advancements in antenna design for handheld RFID readers. Such mobile devices can acquire their locations outdoors using GPS-based positioning with typical accuracies of 1-3 meters and indoors using other wireless technologies such as WiFi, Ultra Wide Band (UWB) or RFID with meter-level accuracy.

---

[2] Maintaining security and user privacy is subject to further research.

On a parallel scope, the market awareness of the business values generated by deploying smart building solutions fosters the growth of the smart technologies that can be deployed in all building types, including residential and commercial. The memory spots as limited storage and processing devices [19] [20] that can use WiFi, Bluetooth or Zigbee technology can play a role in such smart building solutions. A lower cost alternative is a semi-passive RFID tag, which has a relatively long read/write range compared with a typical passive RFID tags, on-board processor and data storage. Mobile devices, according to their different wireless communication capabilities, can use compatible memory spots to disseminate location information or location queries.

V. SIMULATIONS AND RESULTS

In this section, we evaluate our scheme and compare its performance with the peer-to-peer pull dissemination strategy as proposed in [17] through extensive simulation experiments using ns-3 [21]. We are interested in two performance metrics: (1) *localization delay* which represents the average time the scheme takes to respond to a location query and (2) *average overhead* which represents the average number of messages the scheme's participants exchange to respond to a location query. In our evaluation, we study the effect of two main parameters; number of *Memory Spots* and number of *Readers*.

A. Simulation Setup

Using ns-3 and based on Graph-Based Mobility Model for Ad hoc networks, we simulate an area of *200m* x *200m* containing *14* points of interest. The points are linked using pathways of *8m* width through which *Readers* are allowed to move using pedestrian speed ranging from 0.75m/sec to 1.25m/sec. During the simulation, *Readers* are allowed to pause for a period of time (say *10sec*) at each point of interest during their tour. Each *Reader* has a reading range *4m* to interrogate surrounding tags and a *30m* reading range to communicate with *Memory Spots*. We deploy abundant number of stationary tags (*1000*) while we consider only a *100* randomly selected tags to be used during generating queries. In addition, we deploy different numbers of *Memory Spots* at points of interest and in the pathways, which have *30m* reading ranges and may be partially overlapped. We execute the simulation experiments under different settings in terms of number of *Readers* and number of *Memory Spots*. During the simulation, we periodically allow each *Reader* to be interested in a random tag and accordingly generate a location query (every *60sec*). In calculating the localization delay, we compute the time it takes for the *Reader* to get a reply and take the average over all generated queries. In calculating the average overhead, we count number of messages exchanged between *Readers* and *Memory Spots* to satisfy a query and take the average over all generated queries. The total simulation time is *5000sec*, all values are averaged over *10* different independent runs with distinct random seeds.

B. Simulation Results

We examine the simulation results for two dissemination strategies; peer-to-peer pull strategy where *Readers* directly communicate with one another to pull location information on demand [17], and our proposed strategy where *Readers* make use of the available *Memory Spots* to disseminate location information with no direct communication.

1) Localization delay

Fig. 4 and Fig. 5 respectively depict the impact of the number of *Readers* and the number of *Memory Spots* on the localization delay while considering *3* different *synchronization* intervals (*60sec, 90sec and 120sec*). As Fig. 4 shows, increasing the number of *Readers* helps our scheme to rapidly disseminate location queries on *Memory Spots* hence answers more queries with less delay. In case of pull strategy, the localization delay is dramatically decreased but on account of average overhead as seen in Fig. 6. However, the better delay for both strategies is for the scenario of more frequent *synchronization* events. Increasing the number of *Memory Spots* slightly affects the localization delay (by an average of only *3%* as shown in Fig. 5). This effect is due to that the travel time of either the location queries or replies may be longer at higher number of *Memory Spots* with less frequent *synchronization*, resulting in increasing the average overhead as explained in Fig. 7.

2) Average overhead

The impact of the number of *Readers* and the number of *Memory Spots* on the average overhead is depicted in Fig. 6 and Fig. 7, respectively. As shown in Fig. 6, increasing number of *Readers* in pull strategy magnifies the average overhead due to broadcasting location queries and replies among *Readers*. While in our scheme, the average overhead is enhanced by an average of 70% even at less frequent *synchronization* when the number of *Readers* is tripled. This enhancement is because more *Readers*, *Memory Spots* are updated in a timely fashion, hence location queries are satisfied with fewer carry and

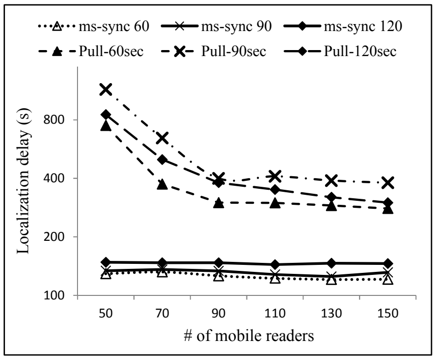


Fig. 4: Effect of # of *Readers* on localization delay (*15 memory spot*).

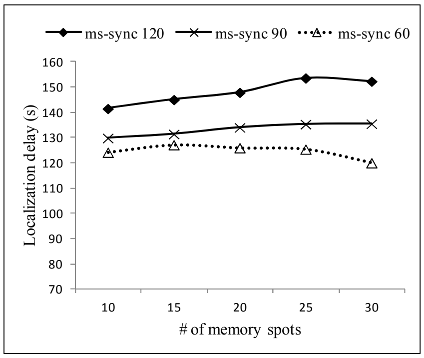


Fig. 5: Effect of # of *Memory Spots* on localization delay (*75 readers*).

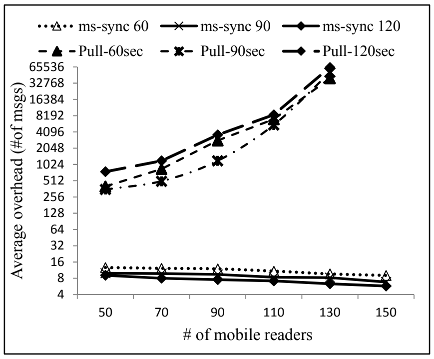


Fig. 6: Effect of # of *Readers* on average overhead (*15 memory spot*).

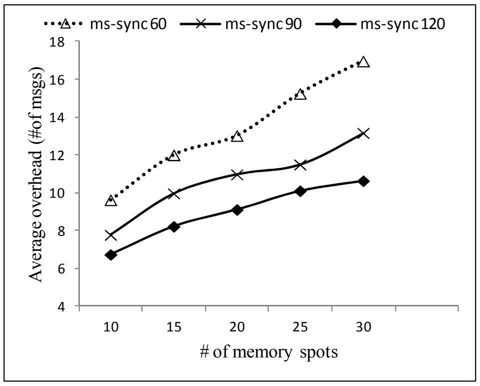


Fig. 7: Effect of # of *Memory Spots* on average overhead (*75 reader*).

forwarding messages. When the number of *Memory Spots* is doubled (see Fig. 7), location queries and replies are required to traverse more *Memory Spots*, which generates more messages and increases the average overhead accordingly. The better average overhead is for the less frequent *synchronization* where *Readers* maintain their location information and carried queries for longer time before updating *Memory Spots*.

VI. CONCLUSION

In this paper, we propose a distributed localization and location information dissemination scheme for typical IoT large-scale dynamic scenarios. Our scheme uses heterogeneous uncoordinated mobile RFID readers to localize mobile or stationary passive RFID tagged-objects and provides high location information availability by means of "memory spots". Memory spots have limited storage and processing capabilities and can use WiFi, Bluetooth, Zigbee or RFID technology to disseminate the locations of passive RFID-tagged objects, and to exchange location queries with lower overhead. The contributions in our approach are that we: (1) employ heterogeneous uncoordinated mobile RFID readers to localize surrounding tags, (2) use memory spots to maintain location information availability and (3) design a distributed protocol for location information dissemination amongst RFID readers through the memory spots. We evaluate our proposed scheme and compare its performance with typical peer-to-peer pull dissemination strategy through extensive simulation experiments using ns-3. Results show that in high dynamic environments, using the pull strategy is infeasible due to its tremendous overhead which inspired us to propose the usage of memory spots as a cost effective enabling infrastructure for distributed location information dissemination. In addition, we concluded that when the number of memory spots increases, generating more coverage overlap, the average overhead is more likely to be high. We plan to investigate the parameters that control the optimal deployment of memory spots in a given environment and application. In our work, we assume that memory spots can only communicate with mobile readers, we plan to investigate the case where memory spots can communicate with one another.

VII. ACKNOWLEDGMENT

This work was made possible by NPRP grant # NPRP4-553-2-210 from the Qatar National Research Fund (a member of The Qatar Foundation). This research is also supported in part by the Canadian Natural Science and Engineering Research Council (NSERC). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.
[2] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of Things, " Internet Computing, IEEE, vol. 14, no. 1, pp. 44-51, 2009.
[3] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges,"Ad Hoc Networks, vol. 10, no. 7, pp. 1497 – 1516, 2012.
[4] D. Zhang, F. Xia, Z. Yang, L. Yao and W. Zhao, "Localization Technologies for Indoor Human Tracking,", International Conference on Future Information Technology, Vol. 5, in press, May 2010.
[5] Khoo, B., "RFID- from Tracking to the Internet of Things: A Review of Developments," Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom) , vol., no., pp.533,538, 18-20 Dec. 2010.
[6] EPC Global, Class 1 Generation 2 UHF air interface protocol standard v.2.0.0, http://www.epcglobalinc.org, 2013.
[7] Z. Junyi, and S. Jing, RFID localization algorithms and applications—a review, Journal of Intelligent Manufacturing, Vol. 20, pp. 695−707, 2009.
[8] I. Farris, A. Iera, and S. C. Spinella, "A novel paradigm to exchange data in RFID piconets," IEEE International Conference on RFID, vol., no., pp.215,222, 2013.
[9] Wei Cheng; Shengling Wang; Xiuzhen Cheng, "Virtual track: applications and challenges of the RFID system on roads," *Network, IEEE* , vol.28, no.1, pp.42,47, January-February 2014
[10] L. Eslim, H.S. Hassanein, W. Ibrahim, and A. Alma'aitah, "Time-Shifted Multilateration for Mobile Object Localization using RFID Crowdsourcing", Technical Report 2013-615, Nov. 2013.
[11] L. M. Ni , Y. Liu , Y. C. Lau and A. P. Patil "LANDMARC: Indoor location sensing using active RFID", Wireless Netw., vol. 10, no. 6, pp.701 -710, 2004.
[12] G. Jin, X. Lu, and M. S. Park, "An Indoor Localization Mechanism Using Active RFID Tag," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing , vol. 1, pp. 40–43, 2006.
[13] A. Bekkali, H. Sanson, and M. Matsumoto. RFID indoor positioning based on probabilistic RFID map and kalman filtering. In Proc. Of WiMOB, 2007.
[14] Y. Zhao, Y. Liu, and L.M. Ni. VIRE: Active RFID-based localization using virtual reference elimination. In Proc. of ICPP, 2007.
[15] M. Bouet and G. Pujolle. L-virt: Range-free 3-d localization of rfid tags based on topological constraints. Computer Communications, 32:1485-1494, 2009.
[16] C. Hekimian-Williams, B. Grant, and P. Kumar, "Accurate localization of RFID tags using phase difference," in Proceedings of the IEEE International Conference on RFID, 2010.
[17] L. Eslim, W. Ibrahim and H.S. Hassanein, "GOSSIPY: A Distributed Localization System for Internet of Things using RFID Technology" Global Communications Conference (GLOBECOM), 2013 IEEE, pp.162-167, 2013.
[18] A. Almaaitah, K. Ali, H.S. Hassanein, M. Ibnkahla, "3D Passive Tag Localization Schemes for Indoor RFID Applications," Communications (ICC), 2010 IEEE International Conference on , vol., no., pp.1,5, 23-27 May 2010.
[19] http://www.st.com/web/catalog/mmc/FM76/CL1766/SC1412/PF219471?WT.ac=fp_jun11 [online] Last accessed Mar. 7, 2014.
[20] http://8devices.com/ [online] Last accessed Mar. 7, 2014.
[21] The network simulator ns-3 Available: http://www.nsnam.org/. Last accessed Jan 7, 2014.