Reciprocal Public Sensing for Integrated RFID-Sensor Networks

Fadi M. Al-Turjman[1], Ashraf E. Al-Fagih[2], Waleed M. Alsalih[3] and Hossam S. Hassanein[1]

[1]School of Computing
Queen's University, Kingston, Ontario, K7L 3N6, Canada
{fadi | hossam}@cs.queensu.ca
[2]Information & Computer Science Department
King Fahd University of Petroleum and Minerals, Dhahran 32161, Saudi Arabia
alfagih@kfupm.edu.sa
[3]Department of Computer Science
King Saud University, Riyadh 11543, Saudi Arabia
wsalih@ksu.edu.sa

***Abstract*—Public sensing is an application in which sensory systems embedded in smart devices, vehicles, residential and public spaces form a collective cloud of data sources from which multi-owned access points realize end-users' service requests. This conception can be further extended under the umbrella of integrated RFID-Sensor Networks (RSNs) to include RFID systems. Such a configuration is heterogeneous by nature and faces many challenges in terms of data delivery and resource management. In this paper, we represent a Reciprocal Public Sensing (RPS) scheme for integrated RSN architectures. Our scheme incorporates heuristic solutions for static sensors and mobile data collectors, in addition to a reciprocal agreement for data exchange over the tiers of the proposed architecture adhering to the social welfare of the network as a whole. We provide simulation results showing how RPS outperforms other data delivery schemes in terms of minimizing delay, packet loss, and energy consumption, in addition to prolonging the overall network lifetime.**

Keywords—RFID; WSN; delay; lifetime; reciprocal agreement.

I. INTRODUCTION

Nowadays, public sensing (PS) influences numerous research and application disciplines utilizing networks of sensory and smart devices. For instance, real-time data on air pollution allow pedestrians (users) to decide on best walking routs. Tagged passengers on metro platforms could be detected, as well, and guided towards their next connection sites. One of the forefront technologies driving this PS vision is RFID [1]. RFID systems, consisting basically of readers and tags, are favored because of their small size and low cost, in addition to their lifetimes that are not limited by battery duration. However, RFID is not solely sufficient to deliver the PS vision since other emerging applications require information beyond the identity or location of an object. For the most part, such applications rely on Wireless Sensor Networks (WSNs). In fact, WSNs technology is directly contributing to the development of PS [2]. The integration of RFIDs and WSNs in RFID-Sensor Networks (RSNs) will enhance their joint capabilities. An RSN node may draw its energy from the electromagnetic field of the reader, which reduces the power consumption to a minimum and extends its lifetime significantly. In addition, RFID tagging provides a wider range than the traditional MAC addressing. Moreover, the RFID tagging enables WSNs to track objects that otherwise are difficult to sense and locate. WSNs, on the other hand, add intelligence to RFIDs by providing detailed information on external parameters such as temperature, pressure, etc. In addition to the ability to inter-communicate over multi-hops regardless of the reader's range of interrogation. Such features are necessary before RFID systems can be used in full-fledged event monitoring [3]. For the aforementioned mutual advantages, RSNs represent a highly appealing solution for a multitude of functional, scalable and cost-effective data collecting solutions. The integration of RFIDs and WSNs will enable a plethora of new applications in the PS framework through supporting the sensing, computing, and communication capabilities into otherwise passive systems.

In this paper, we introduce a Reciprocal Public Sensing (RPS) approach for RSNs. Our scheme responds to data requests from access points by examining several types of stationary data sources including sensors and tags in addition to mobile data collectors while adhering to cost and delay constraints. RPS caters as well to the social welfare of the system by observing load balance while seeking to maximize the network's lifetime.

To this end, we summarize our contributions as follows:

- Our RPS scheme incorporates a multi-tier architecture that introduces Relay Nodes (RNs) as stationary entities between the data sources and requesters. RNs organize the data routing between the top (requesters) and bottom (data sources) tiers of the architecture and relieve data sources from data retrieval and transmission loads.
- We use RSN-specific delay and lifetime metrics to predict a Reciprocal Agreement (RA) factor for the exchanged data in a public sensing application.
- Our scheme addresses network delay and prolongs lifetime of the underlying sensors by agreeing on strategic RA factor in which a mobile data collector might be utilized.

The remainder of this paper is organized as follows. Section II surveys related work in terms of integrated RSN architectures and data delivery schemes. Section III describes our system models and problem statement. Next, our RPS scheme is proposed in Section IV. Section V evaluates the performance results of RPS in comparison to other data delivery schemes. Finally, Section VI concludes this work.

978-1-4673-2480-9/13/$31.00 ©2013 IEEE

II. BACKGROUND

In this section, we outline common RSN integration approaches. In addition, we overview predominant data delivery schemes in Ad hoc networks which resemble RSNs in many aspects including node heterogeneity, mobility and the infrastructure-less nature.

A. RSN Architectures

RSN architectures may follow a variety of integration approaches that can be classified into four main categories: 1) Sensor-Tag (ST), 2) Sensor-Reader (SR), 3) Reader-Relay (RR), and 4) the Mixed (Mix) integration. In ST integration, sensing capabilities are added to RFID tags. For example, authors in [1] proposed a duty cycling solution that attaches an RFID tag to each sensor node. However, this architecture suffers from doubling the load on the integrated node which is required to run two wireless protocols in addition to some aggregation method to overcome the short communication ranges of relaying sensors. This will increase the system's operational and design complexity in large-scale deployment. Contrarily, there was an attempt in [4] to introduce a prototype for large-scale asset tracking that integrates RFID readers with sensor nodes (SR integration). However, integrating RFID readers with sensors is not efficient from the perspective of cost, since RFID readers are to be deployed as densely as regular sensors are. In addition to the limited sensing ranges and power resources of sensor nodes. Accordingly, we proposed in a previous work [5] a variant level of integration incorporating RFID reader and relays together in one Reader-Relay (RR) entity. This design limits the cost complexity in a single component of the architecture while committing data reporting nodes (i.e. tags and sensors) to that sole task without any relaying or processing load. In addition to the RR approach, there is also the mixed (MIX) integration that allows ST nodes to coexist in the same network with integrated readers and relays. For example, authors in [6] proposed an architecture in which gateways are integrated with RFID readers, while sensors are integrated with tags. Yet, this approach suffers from the drawbacks previously mentioned regarding ST integration.

In this article, we build upon our previous work in [5] to propose a cost-effective data delivery hierarchy for large scale RSNs that takes into consideration other factors, viz.; sensors lifetime, and data delivery delay. In the following, we review the most competitive data delivery schemes in infrastructure-less settings.

B. Data Delivery Schemes

In the literature, Data delivery schemes can be classified into offline [7], or online [8] approaches. In offline approaches, delivery plans are pre-set before the system is operational. Any updates on the presumed paths between the nodes require a re-programming of the whole delivery scheme. Meanwhile, the online approaches are less deterministic and more flexible to topology changes in a timely manner. Hence, we focus here on online schemes, as offline approaches are relying mostly on extended amounts of time complexities that render such approaches unrealistic for dynamic topologies targeted in this paper. Conducting a reliable and timely online data delivery in highly dynamic networks is of an utmost importance. Traditional topology-based MANET delivery protocols, such as DSDV, DSR and AODV [9], rely on predetermination of end-to-end routes prior to data transmission. Thus, they are vulnerable to mobility. Moreover, if the path breaks, non-delay-tolerant data packets will be dropped causing a severe deterioration in delivery rates. Geographic routing [10], on the other hand, utilizes location information to forward data packets in a hop-by-hop routing fashion. Next hops are selected according to a greedy approach that measures the largest positive progress towards the destination. However, this approach is sensitive to the inaccuracy of location information. Another problem that faces dynamic routing approaches is the redundant delivery. Broadcasts cause multiple receptions of the same source-generated packet. Yet, since retransmissions are often used as backups to enhance the system's robustness, many opportunistic routing schemes adopt multicast-routing strategies. Nevertheless, such schemes mostly use link-state topology databases to select and prioritize forwarding candidates. This is impractical for highly dynamic mobile environments since it adds to the transmission load by requiring reception of acknowledgments; which are hard to back-trace due to intermittent paths.

Our proposed RPS scheme adopts an opportunistic delay-tolerant routing approach that utilizes the mobility of a few nodes known to be targeting a given destination. This approach does not require any extensive exchange for routing tables nor does it consider an end-to-end path details as long as these mobile nodes are affordable by the RNs. Also, it differs from other PS models in considering the data's sensitivity to delay and to predict the RA value accordingly.

III. SYSTEM MODELS & PROBLEM STATEMENT

In this section we describe the targeted network architecture, in addition to the delay, lifetime and reciprocal data exchange models which form the basis of our RPS scheme. We then present our targeted problem statement in this paper.

A. Network Architecture

In this paper, we consider a two-tier architecture for RSN-based public sensing system. As depicted in Fig. 1, the network is composed of an Access Point (AP), M Relay Nodes (RNs), and N Light Nodes (LNs) deployed in an area of interest. Each AP owns a set of RNs composed of integrated relays and RFID readers. In addition to serving as stationary relays, RNs act as moderators; seeking data on behalf of APs according to the reciprocal agreement factor. Several Mobile Data Collectors (MDCs) patrol the sensing area each moving along a predefined trajectory that overlaps with the transmission range of at least one RN. Without the loss of generality, we assume a circular predefined trajectory for simplicity in calculations. However, other shapes can be easily adopted as well. LNs whose transmission ranges overlap with the trajectory of the MDC are called Boundary Nodes (BNs). In other words, BNs are the LNs that are able to communicate with the MDC directly. Each LN senses the surrounding environment and delivers data to a RN either by multi-hop transmissions through other LNs or by multi-hop transmissions to a BN, which stores the data until an MDC passes by. Meanwhile, the AP directs the operation of the system by:

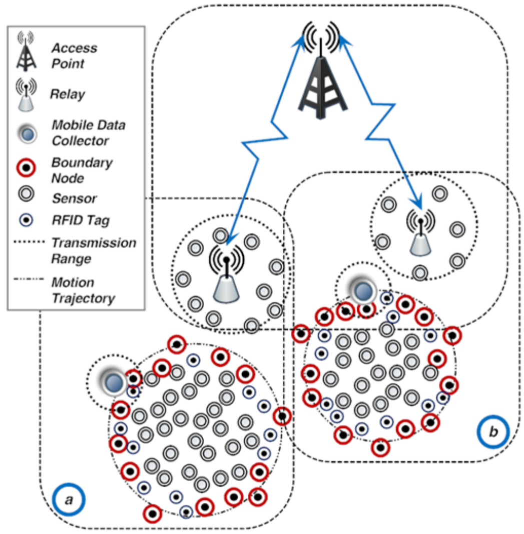

Fig. 1. A two-tier network architecture for RSN in public sensing.

1) Choosing the RNs to be used for data gathering.
2) Deciding on whether data is delivered directly to the RN or through a MDC.

Accordingly, we have two possible scenarios for data gathering at the bottom tier of the network (see Fig. 2):

1) Data delivery to a stationary data collector (i.e. the RN).
2) Data delivery to a MDC that moves along a predefined trajectory. Data packets are forwarded to BNs, where they wait for the MDC to come and pick them up. The MDC then delivers the data to the RN when it enters its transmission range.

B. Lifetme and Delay Models

We state the following assumptions based on the aforementioned architectural model:

1) The transmission range of all LNs is fixed to r meters (i.e. each LN has a transmission disk of radius r *meters*).
2) RNs are stationary and cover all LNs (i.e., each LN has at least one multi-hop path to at least one RN) as discussed in [9].
3) MDCs move along trajectories that are known to the AP and RNs.

In a previous work, we compared these two scenarios in terms of lifetime and delay [11]. We showed that the lifetime of a network with a MDC is determined by the lifetime of BNs as they are the first nodes to run out of energy. Assuming a perfect load balancing over the BNs, we showed that the lifetime of the network with a MDC is:

$$\left\lfloor \frac{E_{init}(2Rr-r^2)}{DR^2E_{tr}} \right\rfloor \text{time units,} \tag{1}$$

where E_{init} is the initial amount of energy each node in the network starts with, the sensing field is a circle of radius R, the transmission range of all light nodes is fixed to r m, each sensor node generates D data packets/time unit, and E_{tr} is the amount of energy units consumed to send one data packet. On the other hand, we showed that the lifetime of a network with a single stationary data collector placed at the center of the sensing field is:

$$\left\lfloor \frac{E_{init}r^2}{DR^2E_{tr}} \right\rfloor \text{time units,} \tag{2}$$

From this comparison, the lifetime of a network with a MDC is longer than that of a network with a stationary data collector by a factor of $\frac{2R-r}{r}$.

We also compared in [11] these two scenarios in terms of delay. For a network with a stationary data collector (Fig. 2-a), the delay a packet encounters is proportional to the number of hops between the packet's source and the stationary data collector. The number of hops between two points can be approximated to be a linear function of the Euclidean distance between them. Therefore, the maximum delay a packet encounters is linearly proportional to the diameter of the network. This results in a delay of αR time units, where α is a constant and R is the maximum distance between the stationary data collector and a LN. On the other hand, the maximum delay in a network with a MDC (Fig. 2-b) is significantly worse. With a MDC, the worst case occurs when a packet arrives to a BN which has just been left by the MDC and it is the first node facing the MDC when it exits the communication range of a RN; such a packet needs to wait for the MDC to complete two full rounds over its trajectory which depends on the speed of the MDC. Let D_{max} denote the maximum delay a packet may encounter waiting for a MDC at some BN in the network. Apparently D_{max} is expected to be much greater than αR.

The amount of energy consumed to send a packet over a multi-hop path is also proportional to the number of hops and, hence, can be approximated to a linear function of the Euclidean distance between the source and the destination. Therefore, the maximum amount of energy consumed to send a packet to a stationary data collector is δR time units, where δ is a constant. On the other hand, the maximum amount of energy consumed to send a packet to a MDC is δR^* time units, where R^* is the maximum distance between a LN and its nearest BN. R^* is expected to be smaller than R.

To summarize, having an MDC has a great potential to prolong the lifetime of the network and save energy, yet it suffers a longer delay as compared with a stationary data collector. This brings together a group of competing objectives and makes a demand for a reciprocal optimization scheme to choose the right data gathering strategy, and this is the main motivation for this work.

C. Reciprocal Agreement Model

We propose a generic decentralized approach that can be tuned to meet different objectives where each RN can belong to a different organization and decides on its service/data Reciprocal Agreement factor (RA_{RN}); a tradeoff of relaying for a monetary value based on the availability of its own resources. We specify three main parameters for each RN to announce its (RA_{RN}) in reply to an AP's request:

- *Delay (D)*: a combination of the time the RN will need to transmit a ready data (T_t) and the time the sensed data will need to arrive to the RN (T_a) such that:

$$D = T_t + T_a \tag{3}$$

Hence, we define a normalized $T_a{}'$ as:

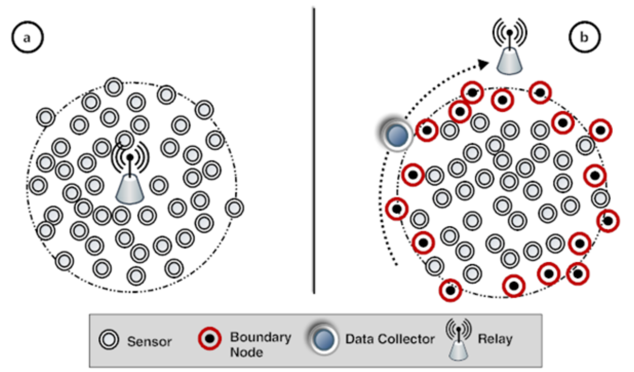

Fig. 2. A network model with (a) stationary relay and (b) with a mobile data collector.

$$D' = \frac{D}{D_{max}} \tag{4}$$

- *Relaying capacity (β)*: Our delivery scheme adopts a data delivery approach where relay nodes have a limited capacity for the maximum amount of data that can be relayed over a specific time period. The RN's capacity is inversely related to its reciprocal (RA) factor. We define a normalized relaying capacity for the set of RNs as:

$$\beta' = \frac{\beta}{\beta_{max}} \tag{5}$$

where β_{max} is the maximum expected capacity.

- *Lifetime (L):* We adopt the general energy consumption model proposed in [11] to evaluate the discussed network lifetime in Eqs. 1 and 2. Thus, we define normalized L':

$$L' = \frac{L}{L_{max}} \tag{6}$$

Based on the three aforementioned parameters, we propose a reciprocal agreement function for each RN such that each packet will be assigned an agreement factor RA_{RN} directly proportional to these parameters:

$$RA_{RN} \propto \frac{\beta' \, L'}{D'} \tag{7}$$

Hence, an AP will choose an RN with the highest RA_{RN}.

D. Problem Statement

According to the abovementioned RSN system model, we address two problems:

1) Finding a generic reciprocal mechanism that manages the relationship between the AP and the RNs. This mechanism is supposed to help the AP to pick the right RN to be used for each data gathering task according to a particular RA factor.
2) Finding the routing infrastructure supporting the delivery of both delay-tolerant data and delay-sensitive data to the AP.

Henceforth, the targeted *RPS* scheme aims towards the following:

Given an abundance of multi-owned APs, RNs, and MDCs, APs are to find the best RNs to cater for their data requests within a specific geographical vicinity while considering the data's RA factor.

IV. RPS Heuristics

Based on the above problem statement, we assume that RNs are stationary and optimally deployed to cover all sensors and tags in its vicinity as discussed in [7]. Each AP will periodically send a beacon announcing its location and the requested data packet(s) with an RA threshold (RA_{TH}) in search for any RN within its interrogation zone. Accordingly, an RN will acknowledge the beacon only if it is willing to participate in data transfer.

Any RN willing to participate in the delivery scheme has to know and announce its RA_{RN} whenever acknowledging an AP's beacon. RA factor is based on an RN underlying network specifications; viz., the residual energy, expected delay, relaying capacity, etc. However, the AP selects its RA_{th} based on its pricing factors and interests, such as the available budget, data value, etc. that are out of the scope of this paper. The final decision of utilizing any RN is left to the AP depending on the highest RA_{RN} it receives. Accordingly, in this section, we introduce the RPS heuristics governing the flow of data between APs and LNs in our RSN architecture according to the aforementioned reciprocal agreement model.

Algorithm 1: Query processing by the AP.

1. **Function AP_Q** (*Q*)
2. **Input**
3. *Q: A query for data to be provided by the AP.*
4. **Output:**
5. *QA: Query answer from an RN.*
6. **Begin**
7. **Initialize** *GV and* RA_{th}. // *GV*: Geographical Vicinity.
8. *Construct the set GVR of RNs in GV.*
9. **Send** *Q* to all RNs in *GVR.*
10. **If** *Ack* is received from a relaying node RN_i **then**
11. Check the received RN's agreement factor *(i.e.,* RA_{RN}*).*
12. **If** ($RA_{RN} \geq RA_{th}$) **then**
13. **Add** RN_i to a set *S* of candidate RNs
14. **End**
15. **End**
16. **Select** RN_k with the highest RA_{RN} in *S*
17. **Assign** Q to RN_k
18. **Receive** *QA* from RN_k
19. **Return** *QA*
20. **End**

Algorithm 1 specifies the steps of a query issued by an AP seeking data for a client's service request from any RN in the set {RN_0, ..., RN_n}. This set is determined according to the geographical vicinity GV of the request (lines 7 and 8). The AP waits then for the RN acknowledgements and bases its selection decision on the RA_{RN} values (lines 9-11), where the chosen RN is selected from a set of candidates as described by line 13. Algorithm 2 details how an RN responds to an AP query. The RN validates the request on two levels. Firs it checks if the requested data is available within the data sources (LNs) under its immediate coverage as depicted by line 8. If not, it then checks with the MDCs within the specified GV. Whenever the data is found, an RN_{RA} value is assigned to it according to Eq. 7 and an acknowledgement is returned to the AP (lines 9-11). Once the AP accepts the offer, the data is forwarded to it, based on lines 14-16. Finally, Algorithm 3 describes the routing scheme executed with delay-tolerant data. This algorithm checks all BNs, where internal LNs use a hop-by-hop communication to reach any of the BNs on the edge of

Algorithm 2: Query processing by an RN.

```
Function RN (Q)
Input:
Q: A data query from the AP.
Output:
QA: Query answer (ignore, Ack including RA_RN, or the Data).
Begin
Initialize GV, and LNs/MDCs in GV.
If Q can be answered by LNs or MDCs then
   Return (QA=Ack) with RA_RN calculated based on Eq. 7
Else
   Ignore
End
If an assignment for Q is received from the AP then
Collect the requested data
QA=requested data.
Send (QA) to the AP
End
```

Algorithm 3: Delay-tolerant routing at an *LN* reporting to RN/MDC.

```
Function LN( )
Begin
for each LN n_i do
   if n_i is a boundary node then
      BN_i.id = i;
      BN_i.next_hop = i;
      BN_i.number_of_hops = 0;
      broadcast BN_i to all neighbors of n_i;
   else
      BN_i.number_of _hops = N + 1;
   end
end
when a LN n_i receives a broadcasted BN_j :
   if BN_j.number_of_hops + 1 < BNi.number_of_hops then
      BNi.number_of_hops = BN_j.number_of_hops + 1;
      BN_i.id = BN_j.id;
      BN_i.next_hop = j;
      broadcast BN_i to all neighbors of n_i ;
   end
end
End
```

the sensing field. Once there, the data waits until it is visited by a MDC so that it can be delivered to an RN. For more elaboration on our RPS approach using, consider the following example.

Example 1. Assume a client has established at some point of time a data query Q, through its nearest AP, as depicted in Fig. 3. Note that each LN-MDC and MDC-RN path has its end-to-end delay characteristics predefined. This also applies to RNs capacity characteristics. Assuming our RN matches the required capacity demand, then the heuristics of RPS selects the path (illustrated in thick red line) that guarantees the best RA_{RN} which must be proportional to the least delay (D_1+D_2) and energy consumption (E_1+E_2) in the selected relay vicinity. Finally, the decision of utilizing this RN relay is then taken based on satisfying the RA_{th} initially set by the requesting AP.

V. Performance Evaluation

In this section, we compare our proposed RPS routing approach against two dominant routing approaches for Ad hoc mobile networks; AODV and NRRA. We chose AODV and NRRA since they represent the two prominent on-demand routing protocols for mobile Ad-hoc networks which are

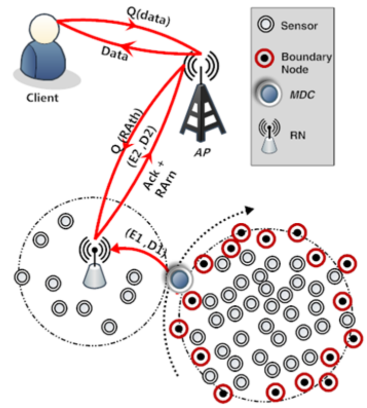

Fig. 3. An example on utilizing the RPS approach.

characterized by are characterized by multi-hop wireless connectivity, frequently changing network topology and the need for efficient dynamic routing protocols. The Ad hoc On-demand Distance Vector (AODV) [8] maintains routes as long as they are needed by the sources. If a source node moves, or a hop on end-to-end route becomes unreachable, route discovery from the source to the destination must be reinitiated if it still requires a route to the corresponding destination. Alternatively, New Reliable Routing Algorithm (NRRA) [13] reduces the number of broken routes. In this algorithm source chooses a stable path for nodes mobility by considering nodes position/velocity information. This algorithm can reduce the number of broken routes efficiently and can improve route stability and network performance effectively. We adopt the energy consumption model proposed in [11] which can be described as follows:

$$E_{Tr}(r, b) = b \times (e_{elec} + e_{amp} \times r_{\gamma}) \quad (8)$$

$$E_{Rc}(b) = b \times e_{elec} \quad (9)$$

where $E_{Tr}(r, _{b})$ is the energy consumed to send b bits over r m, $E_{Rc}(b)$ is the energy consumed to receive b bits, e_{elec} is the energy consumed by the transmitter (receiver) to send (receive) one bit, e_{amp} is the energy consumed by the transmission amplifier for one bit, and γ is the path-loss exponent. In our simulation, r is set to 50 m, e_{elec} is set to 50 *mJ/bit*, e_{amp} is set to 0.1 $mJ/bit/m^2$, and γ is set to 2. The packet size is 512 *bits*. Every sensor node has an initial energy of 50 *J* and generates 150 *pkts/round*. A round is defined as the time span per which all sensors and tags have reported their targeted data. Our simulations involve networks of size 2000, 4000, 6000, 8000, 10000, 12000 and 14000 LNs randomly deployed in a 300×300 m^2, 400×400 m^2, 500×500 m^2, 600×600 m^2, 700×700 m^2, 800×800 m^2 and 900×900 m^2 fields, respectively. Each RN has, for simplicity, a fixed relaying capacity equal to 50% of its generated traffic. For each network size, we test 20 instances and take the average. To generate a trajectory for the data collector, we use a simple method. We divide the sensing field into four equal-size squares; the trajectory of the data collector is a quadrilateral that has a vertex inside each square.

A. Performance Metrics

To compare the performance of the proposed RPS approach, the following four performance metrics are used: 1) Average Delay (*AD*), 2) Average Packet Loss (*APL*), 3)

Average Network Lifetime (*ANL*), and 4) Average Energy Consumed (*AEC*). The *AD* metric is measured in *msec* and is defined as the average amount of time required to deliver a data unit to the AP. *APL*, on the other hand, is the average percentage of transmitted data packets that fail to reach the AP reflecting the bad effects of delay on data delivery over the utilized data delivery approach. The average lifetime (*ANL*) is a measurement of the total rounds the deployed network can stay operational for. It reflects efficiency of the network as well. Finally, *AEC* is measured in *milli-Joules (mJ)* per *byte*. It includes all possible energy consumptions caused by transmission at the LNs, receiving at the BNs, retransmission at the BNs, receiving at the RNs, retransmission at the RNs, and receiving at the APs. Energy may be viewed as a cost factor, as well. We use this metric to show the quality of RPS in choosing better priced solutions. While studying these performance metrics, we vary one main parameter: the total LN count. The LN count reflects the application's complexity and the scalability of the exploited routing scheme.

B. Simulation Results

Figs. 4-7 represent our simulation results comparing delay, packet loss, life time and energy with respect to the number of LNs in the network. LNs where chosen as a direct indicator to the size of the network. As size increases, the ability of the routing scheme to cope with scalability challenges becomes more apparent. Our RSP scheme was compared against two other MANET delivery protocols: AODV and NRRA.

In Fig. 4, we note that average delay across the network increases as its size (i.e., number of LNs) increases. This is to be expected, of course. However, the RPS manages to achieve lower than its competitors. In fact, as the numbers of LNs exceed 10000, the RPS delay plot takes a more stable increase than those of AODV and NRRA. This is due to our delay-tolerant approach that considers routing via BNs and MDCs. The same static/dynamic routing approach explains the superb performance of PRS regarding average packet loss against network size, as shown in Fig. 5. In its worst case, RPS achieves a packet loss that is 300% better than those of AODV or NRRA. Figs. 6 and 7 show the network's reaction in terms of Average Network Lifetime (ANL) and Average Energy Consumed (AEC), respectively, as its size increases. It is apparent from the plots that RPS demonstrates substantially better ANL and AEC than its competitors. In Fig. 6, RPS maintains a higher average life time regardless of the network's size. Even with N = 14,000 LNs, RPS average lifetime is significantly higher than that of AODV or NRRA. In Fig. 7, RPS shows a 60% improvement in AEC over that of AODV for the same value of $N = 14{,}000$. This is directly related to our data collecting strategy that caters to both static (RN) and mobile data collectors as fitting to delay or life-time constraints (Eqs. 7 and 8, respectively).

VI. Conclusion

In this paper, we introduce RPS, a Reciprocal Public Sensing approach for RSNs. Our scheme is based on a two-tier RSN architecture where access points and relay nodes seek data from static sensors/tags possibly via mobile data collectors. Data exchange obeys a reciprocal agreement that caters to the social welfare of the system by observing, delay, load balance and maximizing the network's lifetime. We compare our RPS scheme to other dominant MANET protocols such as AODV and NRRA in terms of delay, energy consumption, lifetime and delivery rate. Our simulation results show that RPS copes superiorly with scalability demands and outperforms its rivals under rapid network size expansion.

References

[1] A. G. Ruzzelli, R. Jurdak, and G.M.P. O'Hare, "On the RFID wake-up impulse for multi-hop sensor networks", *1st ACM Workshop on Convergence of RFID and Wireless Sensor Networks and their Applications*, Sydney, Australia, 2007.

[2] A. T. Campbell, et al., "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12-21, Jul/Aug, 2008.

[3] H. Liu, M. Bolic, A. Nayak, and I. Stojmenovic, "Taxonomy and challenges of the integration of rfid and wireless sensor networks," *IEEE Network*, vol.22, no.6, pp.26-35, Nov/Dec. 2008.

[4] A. Mason, A. Shaw, and A.I. Al-Shamma'a, "Asset tracking: beyond RFID," *7th Annual PG Symposium on the Convergance of Telecommun., Networking and Broadcasting*, Liverpool, UK, June 2006, pp. 267-272.

[5] F. Al-Turjman, A. Al-Fagih and H. Hassanein, "A novel cost-effective architecture and deployment strategy for integrated RFID and WSN systems," *IEEE Int. Conf. on Computing, Networking and Comm.,* Maui, Hawaii, 2012, pp. 835-839.

[6] L. Zhang and Z. Wang, "Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems," *Int. Conf. Grid and Cooperative Comp.,* Hunan, China, 2006, pp.463-469.

[7] F. Al-Turjman, A. Al-Fagih, W. Alsalih and H. Hassanein, "A delay-tolerant framework for integrated RSNs in IoT," *Elsevier's Computer Comm.,* July 2012. (In Press, DOI: 2012.07.001)

[8] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *ACM MobiCom,* Dallas, Texas, 1998, pp. 85-97.

[9] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Elsevier's Ad-hoc Networks*, vol.2, no.1, pp. 1-22, Jan. 2004.

[10] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30-39, 2001.

[11] W. Alsalih, H. Hassanein, and S. Akl, "Routing to a mobile data collector on a predefined trajectory," *IEEE Intl. Conf. on Comm.,* Dresden, Germany, June 2009, pp. 1-5.

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor networks," in *33rd Annual Hawaii International Conference on System Sciences*, Jan. 2000, pp. 1-10.

[13] M. H. Mamoun, "A new reliable routing algorithm for MANET", *Intel Journal of Research and Reviews in Computer Science,* vol. 2, no. 3, June 2011.

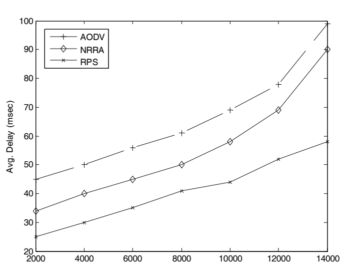

Fig. 4. AD vs. number of LNs.

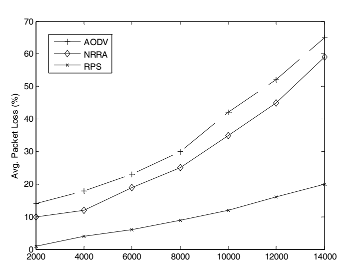

Fig. 5. APL vs. number of LNs.

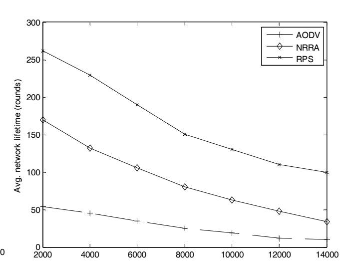

Fig. 6. ANL vs. number of LNs.

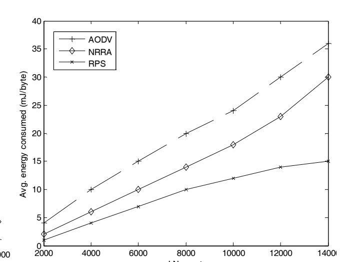

Fig. 7. AEC vs. number of LNs.