Received 1 April 2013; revised 11 July 2013; accepted 15 July 2013. Date of publication 22 August 2013;
date of current version 20 September 2013.

Digital Object Identifier 10.1109/TETC.2013.2278698

A Priced Public Sensing Framework for Heterogeneous IoT Architectures

ASHRAF E. AL-FAGIH[1] (Member, IEEE), FADI M. AL-TURJMAN[2] (Member, IEEE), WALEED M. ALSALIH[3], AND HOSSAM S. HASSANEIN[4] (Senior Member, IEEE)

[1]Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2]School of Engineering, University of Guelph, Guelph, ON N1L 2W1, Canada
[3]Department of Computer Science, King Saud University, Riyadh 11543, Saudi Arabia
[4]School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada

CORRESPONDING AUTHOR: A. E. AL-FAGIH (alfagih@kfupm.edu.sa)

This work was supported by King Fahd University of Petroleum and Minerals, by the National Plan for Science and Technology at King Saud University under Project 11-INF1500-02, and by the Canadian Natural Science and Engineering Research Council (NSERC).

ABSTRACT The proliferation of wireless sensors has given rise to public sensing (PS) as a vibrant data-sharing model. This vision can be extended under the umbrella of the Internet of Things (IoT) to include versatile data sources within smart cities such as cell phones, radio frequency identification tags and sensors on roads, and buildings and living spaces. The facilitation of such a vision faces many challenges in terms of inter operability, resource management, and pricing. In this paper, we present a priced PS framework for IoT architectures. Our framework caters for service-based applications in smart cities where data is provided via a data cloud of multifarious data sources. We propose online heuristics for public data delivery in smart city settings. We also introduce a pricing utility function for data acquisition. Our pricing function considers resource limitations in terms of delay, capacity and lifetime on the data providers' side, as well as user's quality and trust requirements from the requesters' side. We present simulation results showing the efficiency of our scheme as compared with other wireless sensor and mobile *ad-hoc* schemes with respect to scalability, lifetime, delay, delivery ratio, and price.

INDEX TERMS Wireless sensor networks, public sensing, Internet of Things, heuristic algorithms, utility functions.

I. INTRODUCTION

There are billions of sensor-enabled devices that are linked together generating a boundless pool of data [1]. The topologies formed by such devices, including sensors, cellphones, GPS locators, RFID systems and other pervasive objects, collectively build the Internet of Things (IoT); a paradigm that assumes all the aforementioned components to be equally identifiable, tractable and connected [2]. IoT is expected to integrate a multitude of wireless platforms and architectures to provide large-scale information access. One particularly promising model in this regard is Public (or Participatory) Sensing (PS) which employs large-scale sensor networks at low cost by utilizing everyday sensory and mobile devices in applications where data is shared among users for the greater public good [3], [4].

In smart cities [5], under the umbrella of IoT, PS will expand to incorporate heterogeneous data generating/sharing systems including Wireless Sensor Networks (WSNs), database centers, ubiquitous devices, personal and environmental monitoring devices deployed both in metropolitan as well as urban areas. Henceforth, the term "*sensor*" is expanded in this paper to include any form of data source that is either stationary or on transit, regardless of its communication protocol or underlying technology. Hence, this distributed network of sensors will provide a multitude of services to improve the residential experience and quality of living in smart cities. Sensors in such settings are abundant and available with individuals, onboard private and public vehicles and/or deployed on roads and buildings. In such a comprehensive public sensing model, an incentive data

sharing policy is required to motivate sensor owners to participate in the sensing process and to ensure that the provided data is fairly priced. Moreover, the proposed deployment scale introduces challenges regarding the system's limitations in terms of lifetime, available capacity and delay. In addition, quality management policies are to be considered, as well, given the variety of data that is exchanged across the proposed system.

We visualize an IoT-driven PS architecture for smart cities that tackles all the above mentioned concerns. In our proposal, sensors (i.e. data sources), within smaller peripheral networks, populate a broad spectrum of data types, including Voice over IP (VoIP), live and buffered video, identification messages, emergency alerts, geographical and environmental readings into a common data cloud. Each peripheral network is connected to the cloud via a gateway, defined by WiFi hotspots, data servers or even landline hubs. Gateways are responsible for managing the data within their corresponding peripheral networks in terms of provisioning and pricing. Fig. 1 depicts our view of such an architecture where data is generated by masses of data sources and is delivered, possibly via data collectors (static or mobile), to gateways that respond to data requests issued on behalf of the clients by access points that are owned by various service providers.

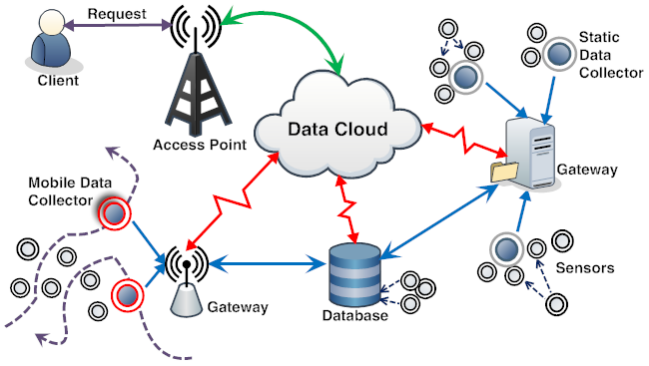

FIGURE 1. Architectural model of IoT public sensing.

To this end, we list our contributions as follows:

- We propose a framework for priced public sensing (PPS) in smart cities based on an IoT architectural model that integrates heterogeneous data sources.
- We categorize data exchanged across the proposed architecture based on its delay and quality requirements. Accordingly, we classify our data delivery schemes into delay-tolerant and delay-sensitive schemes.
- We provide a dynamic two-tier pricing scheme that, from the suppliers' end, adheres to the social welfare of the PS system by incorporating lifetime and capacity constraints while considering, from the users' end, delay, quality and trust metrics to ensure the maximum utility gain.
- We provide heuristics specifying a distributed data delivery scheme that exploits the components of the proposed architectural model, in addition to utilizing our pricing model.

The reminder of this paper is organized as follows: Section II surveys the related work in terms of data delivery and pricing schemes in PS. Section III presents our PPS system models, including the network, lifetime, delay and two-tier pricing models. Section IV defines our problem statement and delivery heuristics. Section V demonstrates our simulation approach and results. Section VI concludes this paper.

II. BACKGROUND

Many research disciplines have been motivated by PS in metropolitan areas [4]–[6]. Accordingly, several PS prototype systems that are based on sensory and Mobile Ad-hoc Networking (MANET) applications have emerged. In this section, we survey some of the prominent PS prototype systems, including their corresponding data delivery and pricing approaches.

A. DATA DELIVERY IN PUBLIC SENSING

The recent explosion of mobile phone market inspired a category of PS prototypes [7]–[10]. These prototypes use sensor-enabled mobile phones to sense their local environments (e.g. pollution and temperature sensing) [7], to monitor their private spaces (e.g. monitoring body vital signals) or to create a binding between tasks and the physical world (e.g. take video or audio samples) [8], [9]. We note that these prototypes are mostly simplistic and dedicated to a single sensory task. MetroSense, which is a more comprehensive PS framework presented in [11], represents a wider vision of a people-centric paradigm for urban sensing. Nevertheless, Metro Sense solely explores sensor-embedded mobile phones to support personal and public sensing.

Joining the capabilities of a multitude of smart devices in a Cloud of Things (CoT) has been proposed in several works [12]–[14]. Other proposals specifically address smart cities applications [15]–[17]. The authors of [15], for instance, highlight how future cities need to collect data from an abundance of low-cost urban sensors including environmental sensors, electric meters, GPS devices and building sensors. The key idea for getting high-quality services from such cheap sensors is the cross-correlation of sensed data from several sensors and analyzing them using sophisticated algorithms. However, none of the aforementioned approaches introduce a comprehensive framework that addresses delivery, resource management and pricing challenges together. Most importantly and from the delivery perspective, they either apply simplistic communication protocols that are either basically cellular (e.g. GPRS or UMTS) coupled with some WiFi or Bluetooth capabilities, or apply algorithms that are intended originally for MANETs such as Ad-hoc On demand Distance Vector Routing (AODV) [18] and Dynamic Source Routing (DSR) [19] which are short-path routing protocols that are based on minimum-hop count. We note that DSR outperforms other protocols in smaller networks of lower load and/or

mobility. AODV, however, outperforms DSR in more stressful situations, with widening performance gaps with increasing stress (e.g. more load, higher mobility). Hence, if a delivery price function is to be proposed for AODV, it would be based on its high control overhead and bandwidth consumption due to its table-exchange-based discovery process, whereas delivery in DSR would be priced according to its performance in highly mobile settings. We, on the other hand, aim at providing a comprehensive pricing function for our framework that considers these and other factors, as will be shown in Section IV.

B. PRICING IN PUBLIC SENSING

Pricing schemes proposed in the literature focus on maximizing the service provider's profit in addition to optimizing the network's resources. From the wider perspective of wireless networks, pricing schemes are classified into flat-rate and parameter-based schemes.

Flat-rate schemes set a fixed price to each connection session regardless of its duration, the type of service requested or the amount of traffic in the network. Examples of common flat rate schemes include metered, packet and Paris-Metro pricing schemes [20]. However, flat-rate schemes have major drawbacks including failing to support users' mobility and often overpricing provided services.

A considerable body of research has been conducted to develop parameter-based schemes [21]–[23] that improve on the flexibility, adaptability and resource allocation mechanisms of their flat-based rivals. Static and dynamic classes of parameter-based pricing schemes are defined according to the scheme's adaptability to wireless technologies and varying user demands. An important aspect related to parameter-based sensing is related to its participatory nature. That is to say, the users' willingness to contribute their data is crucial to the success of the PS system. Hence, incentives are needed to motivate participants.

Numerous incentive-based schemes are proposed, including the work in [24] where groups of mobile users exchange credits. 'Tit for tat' was proposed for peer-to-peer (P2P) networks [25]. In order to download any content, users are required first to contribute their own data. These two incentive-based schemes assume a unified value per data unit and do not consider the quality of the data. In [26], each intermediate node buys a packet from its previous hop with some credits, and then sells the packet to the next hop for more credits. The authors in [27] allow each node to determine the price of its forwarding services based on the availability of its own resources. These previous studies assume that both the source and the destination are charged for exchanging data. However, [28] introduces a virtual credit-based mechanism to collect data and share bandwidth where both the source and the sink (i.e. 3G node) are rewarded for providing and forwarding data. Yet, this framework is limited to vehicular mobile networks. In [29], an envisioned *Internet of Things market* is referred to where the primary asset is the provisioning of sensor data. In such a market, charging individual queries requires a micro payment scheme. Providers, on the other hand, have to increase their future revenues. Accordingly, an IoT-driven pricing scheme has to realize a level of service that aggregates data from several sources, including 4G, WSNs and other sources, to produce better reliable readings. Furthermore, we mark that the aforementioned schemes assume mostly a direct provider-client relation in determining their price mechanisms. However, in large economic systems such as the one we are targeting by our PPS framework, entities known as *intermediaries* are required for coordinating central structures in the IoT market. Intermediaries take care of authentication, billing and interfacing required to find appropriate services within the heterogeneous cloud of sources. We remark that our dynamic pricing scheme and the architecture associated to it address the aforementioned concerns in addition to addressing lifetime, delay and data quality constraints as explained in Section III-D.

III. SYSTEM MODELS

In this section, we explain our network, delay and lifetime models, in addition to our pricing model.

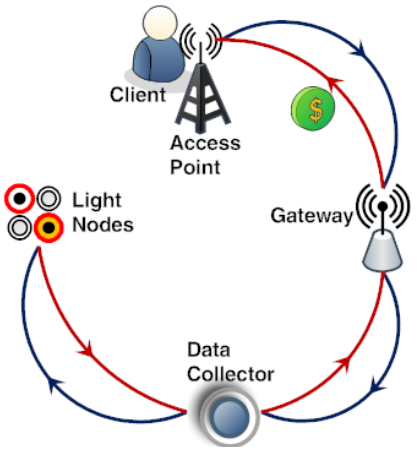

FIGURE 2. **Network model components.**

A. NETWORK MODEL

In this paper, we consider a multi-tier PPS framework with four main components as depicted in Fig. 2. At the top tier of our proposed architecture, Access Points (APs), owned by service providers, initiate data requests based on clients' queries. Data resides at the lower tier of the architecture which includes Light Nodes (LNs) comprising sensors, smart devices, RFID tags and other IoT pervasive entities capable of providing PS data within a binary disk communication range. Each group of LNs is assumed to form a peripheral network according to their deployment and application specifications. Our scheme entirely dedicates LNs to data reading. Thus, data requests and consequent transmissions are not forthrightly conducted between APs and LNs. Each LN delivers its sensed data by multi-hop transmission through other LNs to one or more Data Collector (DC) that could be either station-

ary or mobile. Data collectors are equipped with wireless transceivers or RFID readers and are responsible for forwarding the LNs' data load to gateways (GWs). Gateways are more sophisticated devices which are connected to the cloud via the Internet or any other backhaul. Furthermore, GWs perform an intermediary role by replying to AP data requests in addition to providing them with pricing parameters as will be explained in Section D.

Accordingly, our framework accommodates the following two prominent scenarios:

- A network with a Stationary Data Collector (SDC) located at the center of the sensing field (See Fig. 3-a). Here, all data packets are destined to the DC via multi-hop transmission. Examples of SDCs in smart cities include wireless hotspots and multi-owned static base stations.
- A network with a Mobile Data Collector (MDC) that moves along the boundary (i.e. the perimeter) of the sensing field (See Fig. 3-b). MDCs are represented in smart cities by public transit such as tramway carts and People Movers, equipped with light base stations, that are expected to cover the entire span of their peripheral networks according to a predefined trajectory. We note that our scheme is applicable to fixed trajectories of any shape. Without the loss of generality, we assume at this stage a circular trajectory just to simplify the analysis. Corresponding calculations for other shapes can be derived similarly.

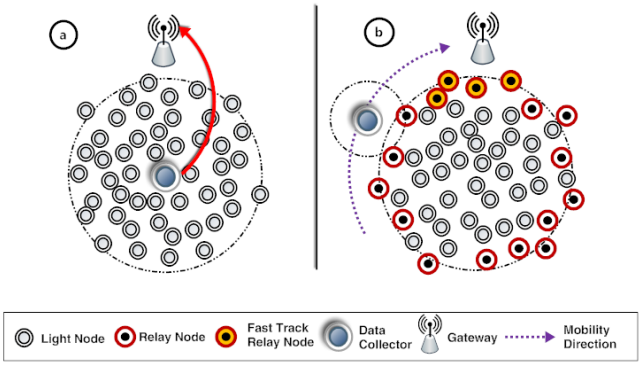

FIGURE 3. Illustration of two data collection scenarios (a) SDC (b) MDC.

As depicted in Fig. 3-b, we label LNs that are able to directly communicate with the DC as Relay Nodes (RNs). In the SDC scenario, RNs are represented by LNs within a single-hop reach of the SDC. This also applies to the MDC scenario where RNs are LNs whose transmission range overlaps with the trajectory of the MDC. In addition, we further label RNs that are the closest to the location of the GW as Fast Track Relay Nodes (FTRNs) as shown in Fig. 3-b. Such nodes are periodically identified by the MDC and are used for more delay-sensitive data as will be shown soon.

We make the following assumptions regarding the lifetime, delay and data delivery models:

- LNs are uniformly distributed over the sensing field.
- The sensing field is a circle of radius Rm.
- The transmission range of an LN is r m.
- All LNs have the same data generation rate where each LN generates M packets per time unit.

For our analysis, we remark though that our proposed framework does not rely on such assumptions – they are made solely to facilitate a quantitative analysis.

B. LIFETIME MODEL

We define the network lifetime as the time until the DC is unreachable (i.e. there exists a LN that does not have a multi-hop path to the DC). In the following, we provide a comparison between the SDC and MDC scenarios that assumes evenly distributed load over RNs (i.e. all RNs relay the same amount of traffic). In addition, we adopt the energy consumption model proposed in [30] which can be described as follows:

$$E_{Tr}(r, B) = b \times (e_{elec} + e_{amp} \times r^{\omega}) \quad (1)$$

$$E_{Rc}(b) = b \times e_{elec} \quad (2)$$

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 to send one bit, e_{amp} is the energy consumed by the transmission amplifier for one bit, and ω is the path-loss exponent. We also assume that every LN starts with an energy supply of E_{init} energy units.

In the following, we compare the two aforementioned scenarios in terms of network lifetime.

1) STATIONARY DATA COLLECTORS

With multi-hop communication, and assuming a circular sensing field, the lifetime of the network is determined by the lifetime of RNs. Since LNs are uniformly distributed over the sensing field, the number of RNs is expressed as:

$$\frac{n\pi r^2}{\pi R^2} = \frac{nr^2}{R^2} \quad (3)$$

where n is the total number of LNs and r is the transmission range of all LNs. These RNs are in charge of delivering all data generated within their designated area to the SDC. Therefore, they transmit Mn packets to the DC. With a perfect load balancing, each RN transmits

$$\frac{MnR^2}{nr^2} = \frac{MR^2}{r^2} \quad (4)$$

packets. Thus, the lifetime of a relaying node is

$$\left\lfloor \frac{E_{init}\, r^2}{MR^2\, E_{tr}} \right\rfloor \quad (5)$$

time units.

2) MOBILE DATA COLLECTORS

Since the DC moves along the perimeter of the sensing field, we have

$$\frac{\pi\left(R^2 - (R-r)^2\right)n}{\pi R^2} = \frac{\left(2Rr - r^2\right)n}{R^2} \quad (6)$$

relaying nodes. With a perfect load balancing and a constant data generation rate, each relaying node will be in charge of transmitting

$$\frac{MnR^2}{\left(2Rr - r^2\right)n} = \frac{MR^2}{\left(2Rr - r^2\right)} \tag{7}$$

packets. Thus, the lifetime of a relaying node in time units is:

$$\left\lfloor \frac{E_{init}\left(2Rr - r^2\right)}{MR^2\, E_{tr}} \right\rfloor \tag{8}$$

From this comparison, we show that the lifetime of a network with an MDC is longer than that of a network with a SDC by a factor of

$$\frac{2R - r}{r} \tag{9}$$

For example, if $r = 100$ m and $R = 1000$ m, the lifetime of a network with a MDC is 19 times longer than the lifetime of the same network with a SDC. We also note that 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 SDC is θR energy units, where θ is a fractional constant. On the other hand, the maximum amount of energy consumed to send a packet to a MDC is θR^* energy units, where R^* is the maximum distance between a LN and its nearest RN. R^* is expected to be smaller than R. We finally note that although the analysis above considers a single MDC, the case for multiple MDCs is straight forward.

C. DELAY MODEL

Delay is defined as the time needed to deliver a packet from a source LN to a destination DC. We remark that we provided in [31] a comparison between the two scenarios in terms of delay. For a network with a SDC (Fig. 3-a), the delay a packet encounters is proportional to the number of hops between the packet's source and the SDC. Therefore, the maximum delay a packet encounters is linearly proportional to the diameter of the network. This results in a delay of ϑR time units, where ϑ is a constant. On the other hand, the maximum delay in a network with a MDC (Fig. 3-b) is significantly worse. With a MDC, the worst case occurs when a packet arrives to a RN which has just been left by the MDC and it is the first node facing the MDC when it exits the communication range of the GW; 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 max_D denote the maximum delay a packet may encounter waiting for a MDC at some RN in the network. It is worth mentioning that max_D is expected to be much greater than ϑR as it depends on the physical motion and speed of the MDC.

Based on the aforementioned lifetime and delay model, we remark that having an MDC has a great potential to prolong the lifetime of the network and save energy, yet it suffers from a longer delay if compared with a SDC. This brings together a group of competing objectives and makes a demand for a scheme to choose the right data gathering strategy, and this is the main motivation for this paper.

D. PRICING MODEL

The public sensing framework we propose is priced. Hence, sensed data in a smart city setting is delivered upon client's request in exchange for a monetary charge or a price to be compensated for by the requesting party. The data provided by LNs is priced according to attributes related to the LN's resource availability such as energy, transmission capacity and the collective lifetime of the peripheral network it belongs to. This latter attribute is decided by the GW connecting the peripheral network to the cloud, which simultaneously acts as an intermediary on behalf of LNs as explained in Section II-B. The requesting party, on the other hand, is to decide on the suitability of the price according to a utility function that conceders the user's service requirements.

We hence provide a two-tier dynamic pricing scheme. At the lower tier, GWs reply to AP data requests with a resource-based service price. At the top tier, the requesting AP generates a utility function based on the user's service requirements, his/her affordable price limit and on the available GW replies to choose from.

1) GATEWAY PRICING

We propose a decentralized approach where each gateway (GW_j) decides on its data/service price (P_{GW_j}); a tradeoff of data for a monetary value based on the availability of resources. We specify three main parameters for each GW_j to announce its P_{GW_i} in reply to an AP_i's request:

- *Delay* (D_{GW}): A combination of the time needed to deliver a packet from a source LN to a destination DC (T_{DC}) and the time (T_{GW}) the sensed data will need to arrive to the GW from the DC such that

$$D_{GW} = T_{DC} + T_{GW}. \tag{10}$$

Hence, we define a normalized D_{GW} as

$$D'_{GW_i} = \frac{D_{GW_i}}{max\_D}, \tag{11}$$

where max_D is the maximum expected delay.

- *Gateway Capacity* (C_{GW}): Our delivery scheme adopts a data delivery approach where GWs have a limited capacity for the maximum amount of data that they can-deliver over a specific time period. The GW's capacity is directly related to its service price (P_{GW}). We define a normalized relaying capacity for the set of GWs as

$$C'_{GW} = \frac{C_{GW_i}}{max\_C}, \tag{12}$$

where max_C is the maximum expected capacity.

- *Lifetime* (L_{GW}): We adopt the general energy consumption model proposed in Section III-B. We define a normalized Lifetime for the set of GWs as

$$L'_{GW} = \frac{L_{GW_i}}{max\_L}, \tag{13}$$

where max_L is the maximum expected lifetime.

Based on the three aforementioned parameters, we propose a price function P_{GW_i} for each GW such that

$$P_{GW_i} = \Phi \frac{C'_{GW_i} L'_{GW_i}}{D'_{GW_i}}, \tag{14}$$

where Φ is a weight variable set here to define a closed form equation for stating a direct proportional relation between price, capacity and lifetime, and an inversely proportional relation between price and delay.

2) ACCESS POINT PRICING

At the top level of our PPS framework, APs are to implement a utility function tailored to cater for the quality, delay and trust parameters of the data type requested by the client. This utility function aims at increasing the users' gain from the coming GW replies according to the user's reservation price (i.e. the maximum price a client is willing to pay) and his/her data requirements. In order to differentiate between the users' requirements of delay and quality in the available data, we categorize the data in smart cities into four main types:

- Hard Real-time bounded Multimedia (HRM)
- Soft Real-time bounded Multimedia (SRM)
- Delay-tolerant data (DTD)
- Delay-sensitive data (DSD)

The four types above cover all data specifications in terms of quality and delay requirements. The first two types (HRM and SRM) are inspired by the work presented in [32] and [33] regarding the type of end-to-end delay and the expected quality guarantees for voice and video data either in real-time (HRM) or buffered (SRM) streams. DTD and DSD are suggested for non-multimedia data (e.g. alert signals, geographic coordinates, environmental measures, etc.) which allows for more relaxed constraints on delay and quality. Nevertheless, some non-multimedia data could have tight delay requirements (e.g. medical alerts). Hence was the distinction between DTD and DSD. Henceforth, we define the following three utility parameters for each data request depending on its type:

- *Delay sensitivity* (DS) of the requested data. The AP may mark some requests to be delay-sensitive or delay-tolerant. And the user may be willing to pay more for less delay, or to compromise some delay for a better (lower) price.
- *Quality of service* (Q) representing the quality of transmission in some predefined aggregate term (e.g. bit rate or average packet loss) where some service (i.e. VoIP and video streaming) require a higher quality level (i.e. quality-sensitive data) in terms of transmission rates as compared to data on traffic updates, for instance. Quality can be also defined with respect of the *reliability* of the source (i.e. in terms of proximity to the sensed phenomenon).
- *Trust factor* (T) is a history function that is calculated at the AP per gateway to represent a GW_j fulfillment measure. A higher T_{GW_j} indicates that previous data exchanges between AP_i and GW_j have been fulfilled according to the parameters (e.g. delay) promised by GW_j. We hence label some data to be trust-sensitive while other data may be trust-tolerant (e.g. data on the weather may be accepted from sources less worthy of trust than is data on a missing person).

Consequently, when a data request ($Data$) is established, the corresponding access point (AP_i) sets a *reservation price* (P_{r_i}) to that request in consistence with the client's service agreement plan, such that P_r defines the maximum price a client is willing to pay for ($Data$), where

$$P_r = \frac{Data_{value}}{Data_{scarcity}}, \tag{15}$$

such that

$$Data_{scarcity} = \frac{no.\ of\ replying\ gateways + \varepsilon}{total\ no.\ of\ gateways}, \tag{16}$$

where ε is a small constant specially placed here to guarantee that $0 < D_{scarcity} \leq 1$, in order to avoid a situation where $P_r = \infty$ in Eq. 15.

As for $Data_{value}$, this parameter is determined according to the quality (Q) and delay (DS) constraints specified by the AP per request according to the data type and the user's service plan. The AP receives the GW reply and if a gateway GW_j replaysback to AP_i, its replywill include the following three parameters:

- The service price (P_{GW_j}) as specified in Eq.14.
- The level of expected delay (D'_{GW_j}), as specified in Eq. 11. The AP will use this parameter after normalization to compute a delay utility component according to the function

$$1 - e^{\frac{-\alpha}{D'_{GW_j}}}, \quad for\ \forall\, GW_j \in \{GW\} \tag{17}$$

where α is a tuning variable chosen to control the shape of the function's curve as will be explained soon.

- The level of quality (Q_{GW_j}) the GW is willing to provide based on its resources. The AP will use this normalized parameter in its quality utility component according to the function

$$\frac{1}{1 + e^{-\epsilon(Q'_{GW_j} - \beta)}}, \tag{18}$$

where ϵ and β are variables chosen to control the shape of the function's curve.

Delay and quality parameters are used, according to Eqs. 17 and 18, to determine $Data_{value}$ to be substituted in Eq. 15 such that:

$$Data_{value} = \left(1 - e^{\frac{-\alpha}{D_{GW_j}}}\right) + \left(\frac{1}{1 + e^{-\epsilon(Q_{GW_j} - \beta)}}\right) \quad (19)$$

In addition, upon receiving a reply, the AP will generate a Trust utility factor (T) per GW. The calculation of T_{GW_j} could follow a function similar to the fuzzy reputation formula presented in [34]. In this paper, we assume an arbitrary value between 0 and 1 to express the trust parameter according to the function

$$(T_{GW_i})^{\gamma}, \quad (20)$$

where $\gamma > 0$ is a weight variable to control the slope of the function by giving more emphasis to the trust parameter.

Eqs. 17, 18 and 20 are used by AP_i to determine the utility score of GW_j after they are tested against the two constraints $D'_{GW_j} \leq DS$ *and* $P_{GW_j} \leq P_r$. Hence, we define our utility function as:

$$U_{GW_i} = \left[\frac{P_r}{P_{GW_i}}\right] \times (T_{GW_i})^{\gamma} \quad (21)$$

The function in Eq. 21 utilizes both prices components P_r and P_{GW_i} and employs delay, quality and trust parameters in a manner that maps the expected user experience to changes in individual utility components.

In order to highlight the individual effects of the aforementioned parameters on the utility function representing our AP pricing model we present in Fig. 4 plots for each of the utility parameters: DS, Q and T according to Eqs. 17, 18 and 20, respectively. In Fig. 4-a, DS is plotted with a constant α that determines the decrease rate of the delay exponential utility component expressed in Eq. 17 as expected delay increases. By varying the value of α, it is possible to achieve different levels of delay-tolerance as shown in Fig. 4-a, where we chose $\alpha = 0.5$ for delay-tolerant data and $\alpha = 0.1$ for more delay-sensitive data. We note that, for a delay-sensitive data request, a very low delay has to be achieved in order to provide a high delay utility component.

For the quality utility component (Fig. 4-b), we adopt the Sigmoid function where the tolerance to quality is expressed by controlling the inflection point denoted by the value of β in Eq. 18. Thus, if the requested data is quality-sensitive (e.g. VoIP), the function will require a higher value of Q before the utility curve increases (as depicted in the lower plot with $\beta = 0.8$ in Fig. 4-b). In contrast, lower constraints on quality require a utility that increases rapidly at a lower value of Q, which can be achieved with an early inflection point ($\beta = 0.5$). The value of ϵ is chosen here to be 10. Lastly, Fig. 4-c shows the plot of the trust function $T \in [0, 1]$ based on the history of GW_j. Note that an AP can give more emphasis to this parameter through the factor γ to particularly penalize GWs with bad service accounts. This is shown in the lower plot of Fig. 4-c ($\gamma = 2$), whereas the upper plot is a result of $\gamma = 1$ indicating less emphasis on trust or data that is not trust-sensitive. We note here that we multiply the Trust component by the rest of the utility function in Eq. 21 to balance the effect of "deceiving" GWs that may offer attractively low P_{GW} in order to pass false quality promises. Moreover, we divide the utility function by P_{GW} in order to protect the client from situations where two or more GWs happen to achieve almost equal utility scores while charging for prices that, although less than P_r, largely vary.

Fig. 5 show a 3-D rendering of the variation of the overall utility calculated by Eq. 21 as the individual utility parameters are varied. The worst utility scores are indicated by the blue color at the lower-left corner of the cube. These worst utilities are associated with extremely low Trust, low Quality and extremely high Delay functions. On the contrary, the best utility scores are indicated by the red color at the top-right corner of the cube. The maximum utility score is associated with extremely high Trust, high Quality and extremely high Delays. When gateways' replies arrive at an AP, several utility scores are generated according to the returned utility parameters of each GW. The AP is responsible of computing the best GW to be chosen based on its individual utility score that is best suitable for the type of data requested and the level of quality specified by the user.

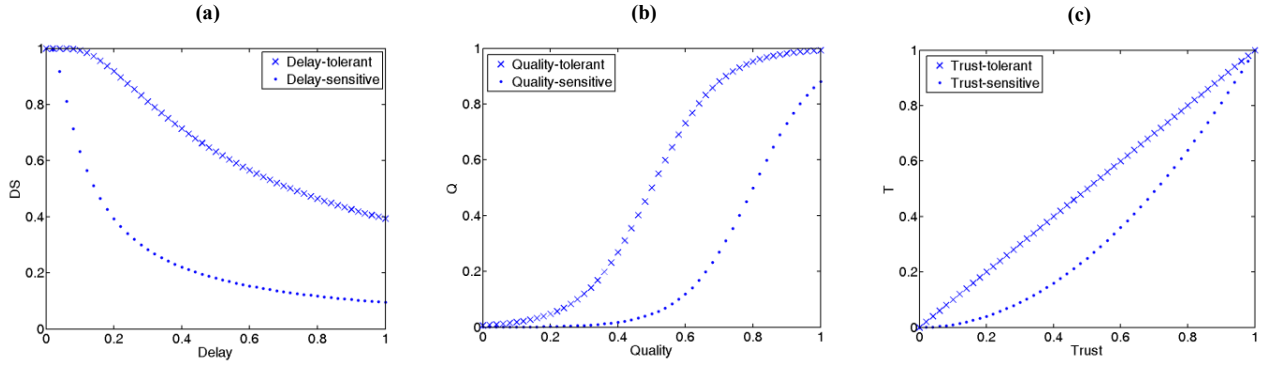

FIGURE 4. Plots for (a) Delay (b) Quality and (c) Trust utility parameters.

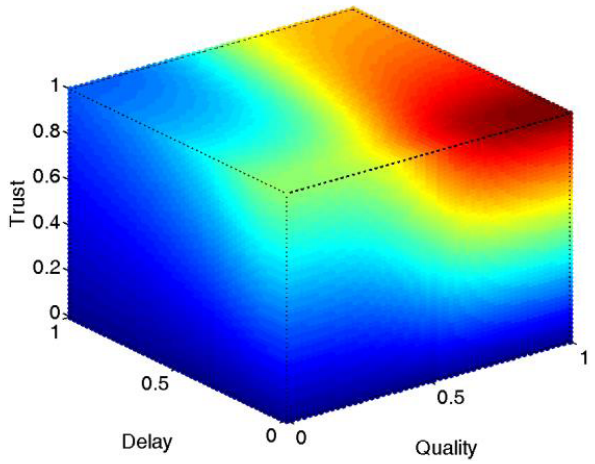

FIGURE 5. **Variation of utility function.**

To demonstrate the impact of the utility parameters on the type of requested data in our PPS scheme, Fig. 6 and Fig. 7 compare the utility scores defined by Eq. 21 for the four data types: HRM, SRM, DSD and DTD while varying delay and quality requirements. We set the following values for each data type:

$$\begin{aligned} \text{HRM}: &\quad \alpha = 0.1, \beta = 0.85 \\ \text{SRM}: &\quad \alpha = 0.5, \beta = 0.6 \\ \text{DSD}: &\quad \alpha = 0.3, \beta = 0.35 \\ \text{DTD}: &\quad \alpha = 0.9, \beta = 0.2. \end{aligned}$$

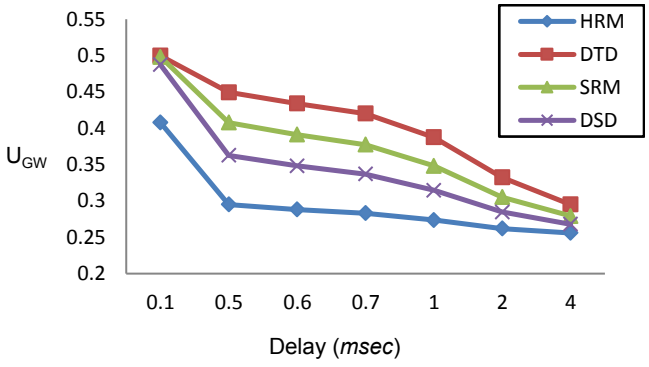

FIGURE 6. **Utility vs. Delay for different data types.**

The values of γ and ϵ are fixed for all data types at 2 and 10, respectively. The values mentioned above were chosen based on the plots in Fig. 4 to reflect the nature of each data type where HRM is the most sensitive in terms of quality and delay, whereas DTD is the most tolerant among the four types in terms of quality and delay requirements. In Fig. 6, we note that the four plots follow the benchmark trend depicted in Fig. 4-a. HRM is the strictest in terms of delay component. The utility plot of HRM deteriorates faster than the rest of data types at $D = 0.5$ msec. DTD is the most tolerant in terms of delay. It achieves a higher utility score throughout the delay axis.

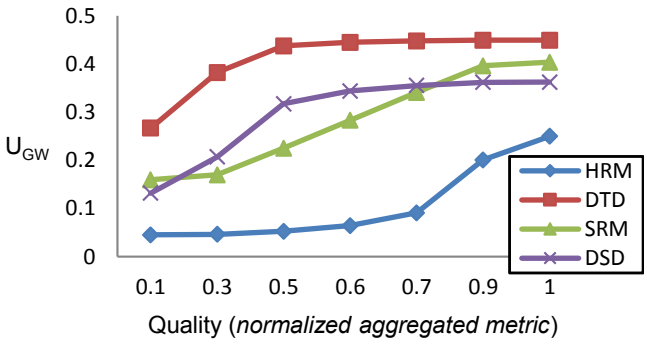

FIGURE 7. **Utility vs. Quality for different data types.**

Fig. 7 shows how different data types react to quality requirements with respect to our utility function. Again, HRM and DTD are impacted the most by quality sensitivity. As for DSD, we see that it is more lenient than SRM, which is represented by buffered video streams, for instance. DSD reaches a utility saturation level faster (at $Q = 0.75$) whereas SRM continuous to demand utility to serve its quality requirement.

IV. PROBLEM STATEMENT & PPS FRAMEWORK

According to the abovementioned PPS system models, the proposed framework addresses two significant problems:

1) Finding a mechanism that manages the relationship between the GWs and the APs at the top-tier of the delivery model. This mechanism is supposed to help each AP to pick the right GW for each data request according to a particular set of utility parameters.
2) Finding the routing infrastructure for the lower-tier of the model that supports delivery of both delay-tolerant and delay-sensitive data to the AP.

Henceforth, the targeted PPS framework aims towards the following:

Given an abundance of multi-owned APs, GWs, and MDCs; APs are to find the best GWs to answer their data requests within specific geographical vicinities while considering different utility parameters for different types of sensed data.

The following two subsections explain our methodology to tackle each of the aforementioned problems.

A. TOP-TIER GATEWAY SELECTION

We specify two algorithms at the level of APs and GWs. These algorithms give details about how an AP will send a data request and how it will process the parameters received from responding GWs into the utility function. The algorithm at the GWs level sets the parameters to be included in a reply to an AP's request and shoes how each GW will determine

Algorithm 1 AP Seeking GW to Provide Priced Data

1. **Function** ***AP (Client_Req.x)***
2. **Input**
3. *Client_Req.x: A client's data request for some service* x *to be provided by this AP.*
4. **Output:**
5. GW_i*: A selected* $GW_i \in \{GW_0, \ldots, GW_n\}$ *to pay and receive data from.*
6. **Begin**
7. **Set** *Data* , *Data.type* //define requested data and its type according to four predefined types
8. **Set** DS , P_r for *Data*
9. **Initialize** *GV, GW{}* //geographic vicinity and set of replying GWs
10. **Request** *Data* **from** *GWs in GV*
11. **While** an *Ack* is received from GW_i
12. **If** $P_{GW_i} \leq P_r$ **and** $D'_{GW_i} \leq DS$ **then**
13. **Add** GW_i to $\{GW\}$
14. **Check** T_{GW_i}
15. **End**
16. **Based on** *Data.type* **do** //service utility preferences
17. **For each** $GW_i \in \{GW_0, \ldots, GW_n\}$ **do**
18. **Calculate** U_{GW_i} //Equation (21)
19. **End**
20. **Select** GW_i with best U_{GW_i}
21. **Pay** P_{GW_i} **to** GW_i
22. **End**

Algorithm 2 GW Reply to AP Query Request

1. **Function** ***GW (Data)***
2. **Input:**
3. *Data: A data request from* AP_i.
4. **Output:**
5. *RA: Request answer that could be an Acknowledgement to the Data request including Price, Quality and Delay parameters.*
6. **Begin**
7. **Initialize** *list of static and dynamic* ***sensors*** *in GV*
8. **If** requested *Data* is available at static sensors **then**
9. **Set** *Delay = 0*
10. **Set** P_{GW} //Equation 14
11. **Return** (*RA=Ack*) with P_{GW}, available Q_{GW}
12. **Else if** the requested data is available at mobile sensors **then**
13. **Set** P_{GW} //according to different *Delay* expectation
14. **Return** (*RA=Ack*) with P_{GW}, available Q_{GW} and expected D_{GW}
15. **Else**
16. **Ignore**
17. **If** this *GW* is selected by AP_i **then**
18. **Return** (*QA=Data*)
19. Charge AP_i with P_{GW}
20. **End**

its service (data) price according to the availability of its resources. Algorithm 1 specifies the steps of a query issued by an AP seeking data (for a client's service request) from any GW in the set $\{GW_0, \ldots, GW_n\}$. This set is determined according to the geographical vicinity (GV) of the request (line 9). As mentioned earlier, the AP waits for the GW acknowledgements and bases its selection decision on the returned utility parameters (lines 12, 13), in addition to P_{GW_i} and T_{GW_i} of each responding GW which is calculated by the AP (line 14). Then, the utility score of each GW is calculated according to the type of data and the required services (lines 16–18). The AP pays the price finally to the chosen GW with a request to release the data (lines 20, 21). Algorithm 2 shows how a GW responds to an AP data request. The GW validates the request at two levels. First it checks if the requested data is available within LNs under its immediate coverage (lines 8–11). If this is the case, the delay component (T_{GW}) in Eq. 10 is expected to be zero and the whole delay factor is hence drastically lower than the following scenarios (lines 12–14) in which the GW checks with MDCs within the specified GV. In either case, whenever the requested data is found, utility parameters of the GW are calculated and included in its acknowledgement to the AP (lines 11 and 14). Once the AP accepts the offer, the data is sold to it (lines 17–19).

B. LOWER-TIER DATA DELIVERY

The lower tier involves Data Collectors (DC's) that collect different types of data from Light Nodes (LN's) and deliver them to GW's. Here, we give a routing infrastructure for the lower-tier of the network that supports delivery of both delay-tolerant and delay-sensitive data to DC's. In order for LNs to deliver their delay-tolerant data to the DCs, they need to have a path to at least one Relay Node (RN). –Remember that a RN is a LN that can communicate directly (in a single-hop) with the DC. To do so, RNs broadcast their identity at the deployment stage and each LN keeps a record of the next hop towards some relaying node. Each LN n_i has a Relaying Node Record (RNR_i) which has the following fields:

- *id*: the id of the relaying node to which delay-tolerant data will be sent.
- *Next_hop*: a neighbor of n_i which is used as a next hop towards the relaying node.
- *Number_of_hops*: the number of hops to the relaying node.

These records are used to determine the route from a LN to a DC. When data is received by a RN, it is stored there until a Mobile DC(MDC) passes by to pick the data up. The data is then carried by the MDC until it becomes within the communication range of a GW; at that point the MDC delivers the data to the GW. Delay is dominated here by the physical motion of the MDC and has two factors. The first is the speed of the MDC. The second is the distance travelled by the MDC between the RN, where the data is picked up, and the GW. While controlling the speed of the MDC is out of the scope of our work, we can minimize the travelled distance by sending the data to an RN that is very close to the GW (i.e. FTRN). Each LN n_i maintains an FTRN Record ($FTRNR_i$) with the same structure as that of RNRs. In consequence, for delivering delay-tolerant and

delay-sensitive data, Algorithm 3 describes the process of setting the FTRNRs and RNRs of all LNs, assuming that each LN uses the nearest RN. Note that this process will construct a tree for each RN; the tree of a RN n_i is rooted at n_i and involves all LNs whose nearest RN is n_i. FTRNRs and RNRs will be identified at the initialization of the network.

V. PERFORMANCE EVALUATION

The proposed sensor environments were simulated using MATLAB. We compare the data delivery scheme of our PPS framework against Ad-hoc On-demand Distance Vector (AODV) and the New Reliable Routing Algorithm (NRRA) [35]. NRRA is a variant of the Dynamic Source Routing (DSR) protocol that uses source routing instead of relying on routing tables in intermediate devices. However, NRRA enhances the link failure in DSR by reducing the number of broken links. NRRA chooses a stable path for nodes mobility by considering nodes position and velocity information. This algorithm can reduce the number of broken routes efficiently and can improve route stability and network performance.

We chose AODV and NRRA since they represent the two prominent on-demand routing protocols for mobile Ad-hoc networks which are characterized by multi-hop wireless connectivity, frequently changing network topology and the need for efficient dynamic routing protocols. NRRA uses source routing and exploits caching aggressively to maintain multiple routes per destination with a notable performance degrade with increasing mobility. AODV, on the other hand, uses routing tables, one route per destination and destination sequence numbers to prevent loops and to determine freshness of routes. Yet, AODV may suffer from heavy control overhead and unnecessary bandwidth consumption due to periodic beaconing. We remark, however, that we employ here modified versions of AODV and NRRA particularly suited for our proposed framework. This is because the original algorithms specify end-to-end delivery routes, while the versions we use here are oblivious on the link state within the data cloud. We define this as the *cloud effect*. In other words, the aforementioned algorithms in their original description are restricted by their corresponding GWs and their view of the topology is limited at their own peripheral networks. Hence, we will refer to our modified versions of AODV and NRRA through the remainder of this section as PODV and PRRA, respectively, where the '*P*' stands here for *Peripheral*. We point out that the original AODV and NRRA approaches are not hierarchical. Thus, we employed the modified versions of them (PODV and PRRA) to make them suitable for our proposed hierarchical framework, where the modified versions take into consideration the in-network nodes heterogeneity and choose next hop based on types of the surrounding node types.

As previously mentioned in Section III-B, we adopt the energy consumption model proposed in [30], which is described in Eqs. 1 and 2. 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 ω 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 4000, 6000, 8000, 10000, 12000 and 14000 LNs randomly deployed in a field of radius R that ranges from 400 km to 1400 km in increments of 100 km. We also simulate environments for a fixed LN count of 4500 while varying the DC count from 5 to 30 in increments of five, with an average speed of 45 *km/h*. 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 DC, we use a simple method. We divide

Algorithm 3 Delay-Tolerant and Fast Track Routing Records

```
Function RN( )
For each LN n_i do
        if n_i is a FTRN then
                FTRNRi.id=i;
                FTRNRi.next hop=i;
                FTRNRi.number of hops=0;
                broadcast FTRNRi to all neighbors of n_i ;
        elseif n_i is a RN then
                RNRi.id=i;
                RNRi.next hop=i;
                RNRi.number of hops=0;
                broadcast RNRi to all neighbors of n_i ;
        else
                RNRi.number of hops = ∞;
        end
end

when a LN n_i receives a broadcasted RNRj :
        if RNRj .number of hops + 1 < RNRi.number of hops then
                RNRi.number of hops = RNRj.number of hops +
        1;
                RNRi.id = RNRj .id;
                RNRi.next hop = j;
                broadcast RNRi to all neighbors of ni ;
        end
end
when a LN n_i receives a broadcasted FTRNRj :
        if FTRNRj .number of hops + 1 < FTRNRi.number of hops
then
                FTRNRi.number of hops = FTRNRj.number of
        hops + 1;
                FTRNRi.id = FTRNRj .id;
                FTRNRi.next hop = j;
                broadcast FTRNRi to all neighbors of n_i;
        end
        if FTRNRj .number of hops + 1 < RNRi.number of hops
then
                RNRi.number of hops = FTRNRj.number of hops
        + 1;
                RNRi.id = FTRNRj .id;
                RNRi.next hop = j;
                broadcast FTRNRi to all neighbors of n_i;
        end
end
```

the sensing field into four equal-size squares; the trajectory of the DC is a quadrilateral that has a vertex inside each square.

As for the arbitrary variables specified in our AP pricing function (Eq. 21) we set $\alpha = 0.5$, $\beta = 0.5$, $\gamma = 2$ and $\epsilon = 10$.

A. PERFORMANCE METRICS

To compare the performance of the proposed PPS approach, the following three performance metrics are used:

1) Average Network Delay (AND), which is measured in msec and is defined as the average amount of time required to deliver a data unit to the AP.
2) Average Packet Delivery (APD), which is set here as a quality measure. It is the average percentage of transmitted data packets that succeed in reaching the AP reflecting the effect of delay on data delivery over the utilized data delivery approach.
3) Average Network Lifetime (ANL), which is a measurement of the total number of rounds the deployed network can stay operational for.

While studying these performance metrics, we vary five main parameters:

1) The size of the network in terms of total LN count. This reflects the application's complexity and the scalability of the exploited routing scheme.
2) DC count while fixing the LN count.
3) Pause time for MDCs as a major delay factor.
4) Sensing field radios (R) per MDC.
5) Average packet generation rate per time round (M) as an indicator of the traffic load across the network.
6) Price at the GW level P_{GW} to observe the influence of its increment on performance in terms of delay and lifetime.

B. SIMULATION RESULTS

We present our simulation results covering performance variations with respect to four primary factors stemming from our utility function (Eq. 21). These factors are: quality level (represented here in APD), delay (seconds), lifetime (rounds) and price (P_{GW}). Figs. 8–18 show our results. In the following subsections, we detail our observations regarding these results.

1) PERFORMANCE WITH RESPECT TO QUALITY

In Fig. 8, we see that the APD decreases for all three delivery schemes as the size of the network increases in terms of LN count. This is expected since a larger network size implies longer paths and higher probabilities for link loss. A larger node count raises the risk of node failure and, hence, dropped packets. Thus, choosing smaller peripheral networks is better for the quality gain according to Eq. 18. The results in Fig. 8 show that PPS substantially outperforms both PODV and PRRA. Several factors contribute to this superior performance of PPS. First, our scheme assumes a heterogeneous environment with abundance of data sources where several copies of the required data may be available. The other schemes suffer from their own disadvantages. PODV relies on multiple history-based routing tables which is inadequate for heavy mobile data traffic. PRRA is a source-routing protocol, which means that a route maintenance mechanism does not locally repair a broken link. In addition, the connection setup delay is higher than that in table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Furthermore, PODV and PRRA are not aware of the link state in the cloud beyond their corresponding GWs.

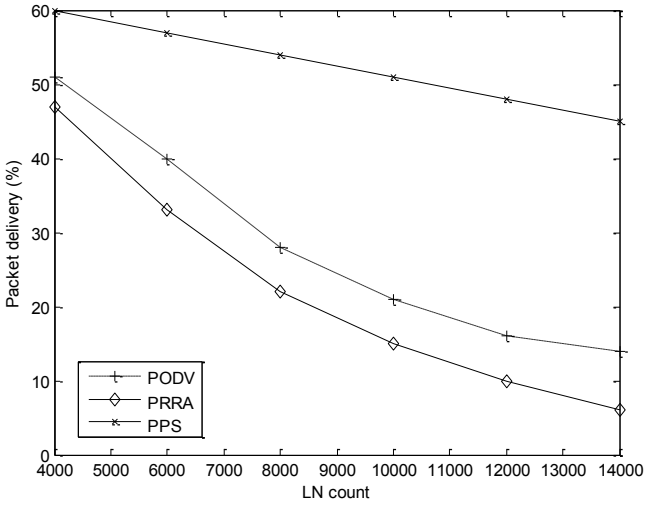

FIGURE 8. Average packet delivery vs. LN count (100 GWs, 5-30 DCs).

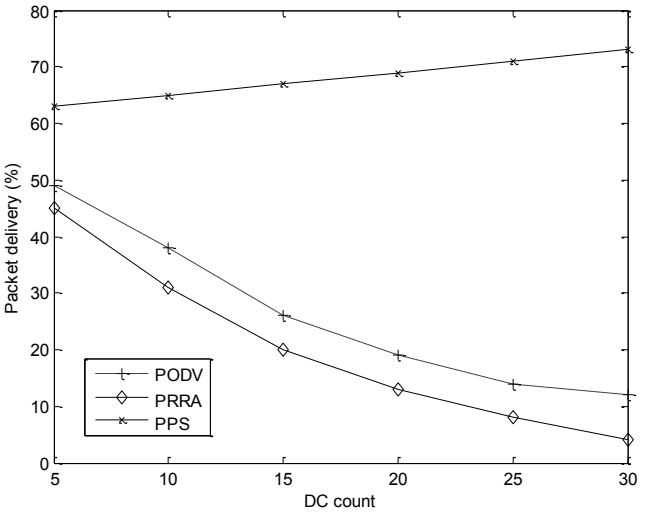

FIGURE 9. Average packet delivery vs. DC count (100 GWs, 4500 LNs).

Fig. 9 examines APD rates with the network size expressed in DC count. We note that while PODV and PRRA show a

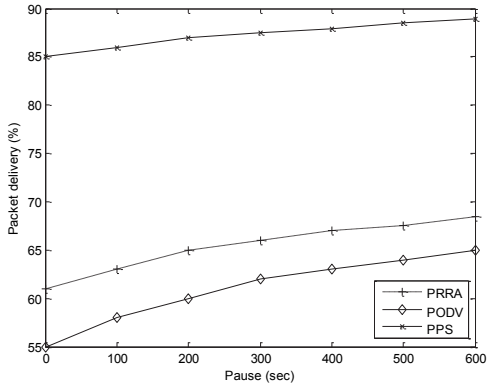

FIGURE 10. Average packet delivery vs. pause (100 GWs, 5-30 DCs, 4500 LNs).

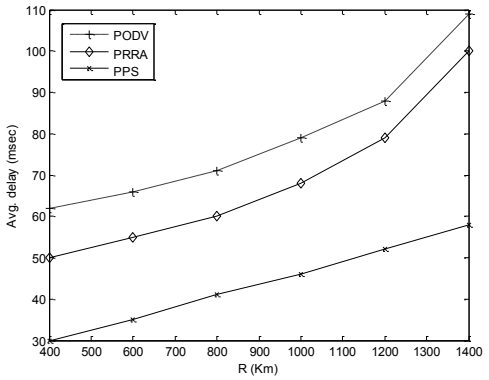

FIGURE 11. Average delay vs. sensing field radius (R) (100 GWs, 5-30 DCs, 4500 LNs).

declining performance almost similar to that of Fig. 8, PPS shows an improvement in delivery as the number of DCs increases. This is because PPS's architecture incorporates in its core the existence of DCs, which gives it an advantage over other delivery schemes. PODV performs slightly better than PRRA since PRRA is challenged by increased mobility, which is subsequent to increases in DCs.

Fig. 10 compares APD against pause time. We see that for all three schemes, the delivery increases slightly as the pause time for mobile nodes increases. This is normal since more pause time ensures a wider window for control and data message exchange, especially for large data loads. Pause time also helps PRRA to discover unknown routes via flooding the network with route requests. However, PPS again shows a better performance as compared to its rivals. In the worst case, PPS is 28% better in delivery than PRRA, which outperforms PODV as mobility decreases (i.e. pause time increases).

2) PERFORMANCE WITH RESPECT TO DELAY

Fig. 11 shows the results of comparing Average Network Delay (AND) with the sensing field radius (R). We see that delay and the size of the sensing field are directly proportional. Yet, because PPS utilizes multiple DCs per sensing field, in addition to adopting an approach that serves delay-sensitivity by defining RNs and FTRNs (Algorithm 3), we note that its delay increaseis steadier and lower than that of its rivals.

The effect of utilizing DCs in PPS is further demonstrated in Fig. 12, where the Average Network Delay (AND) shows a sharp decline unattainable by neither PODV nor PRRA as the DC count increases.

We note that AND is proportional as well to pause time (see Fig. 13). The superior performance of PPS in terms of lower delay is attributed here to its delay-sensitive routing and source selection approach as explained in Algorithms 2 and 3.

3) PERFORMANCE WITH RESPECT TO LIFETIME

In Fig. 14, the increase in generation rate causes PRRA to increase its route discovery process which exhausts the ANL.

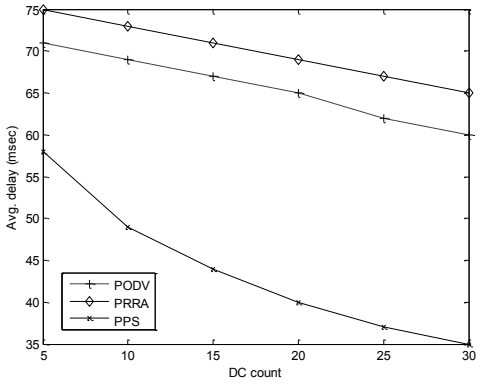

FIGURE 12. Average network delay vs. DC count (100 GWs, 4500 LNs).

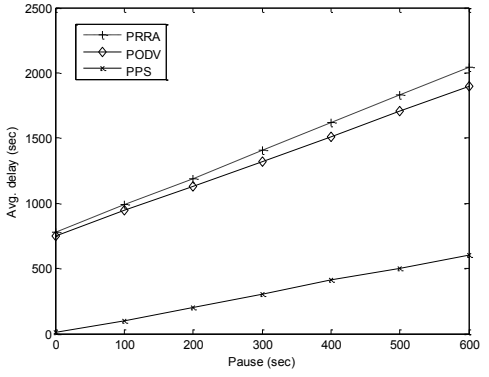

FIGURE 13. Average network delay vs. pause time (100 GWs, 5-30 DCs, 4500 LNs).

In contrast PSS has a higher ANL because its data delivery scheme is top-down. LNs are not exhausted by routing loads. Rather, intermediary GWs are responsible for replying to data requests issued by APs. In addition, our categorization of LNs to RNs and FTRNs has an apparent effect on reducing the transmission load over the collective set of LNs within each peripheral network. Fig. 15 shows that the increase in DC count significantly improves the lifetime of the system under PPS. More DCs facilitate delivery according to PPS algorithms and relieves LNs from further relaying tasks.

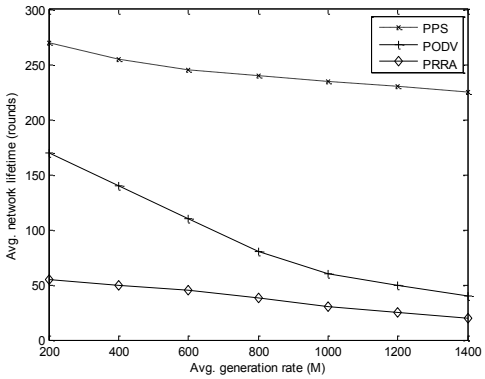

FIGURE 14. Average network lifetime vs. M (pkt/round) (100 GWs, 5-30 DCs, 4500 LNs).

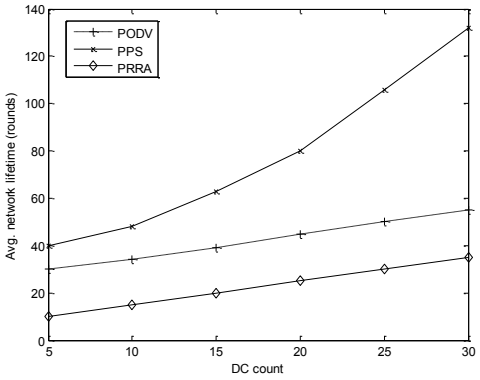

FIGURE 15. Average network lifetime vs. DC count (100 GWs, 4500 LNs).

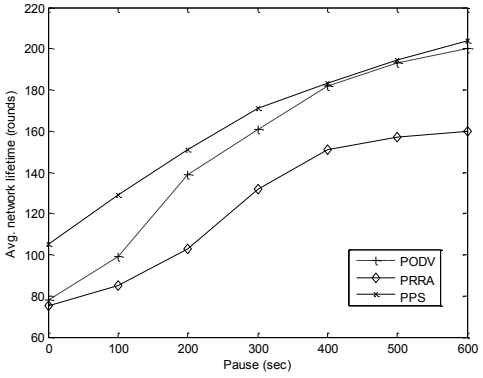

FIGURE 16. Average network lifetime vs. pause time (100 GWs, 5-30 DCs, 4500 LNs).

As for PODV and PRRA, we note from Figs. 14 and 15 that PRRA delivers better performance under low mobility conditions, whereas PODV outperforms PRRA when more MDCs are present. In Fig. 16, we study the effect of MDCs' pause time on ANL, which is a measurement of the total rounds the deployed network can stay operational for. We note that PRRA performs the worst among the three schemes, which agrees with the aforementioned observation on the relation between high node mobility and PRRA.

4) PERFORMANCE WITH RESPECT TO PRICE

As for the influence of data prices set at the GW level, we assign this parameter to GWs in PODV and PRRA by calculating their average performance in terms of the parameters defined in Eq. 14 (i.e. GW capacity, lifetime and delay). The GW price is incremented by some specific value in each round (e.g. 1 price unit per round). Since the selection of GWs is influenced by their announced prices according to our scheme, we apply up to 90% price increments on each GW and remark on the increments' effect on reducing average network delay and lifetime, respectively.

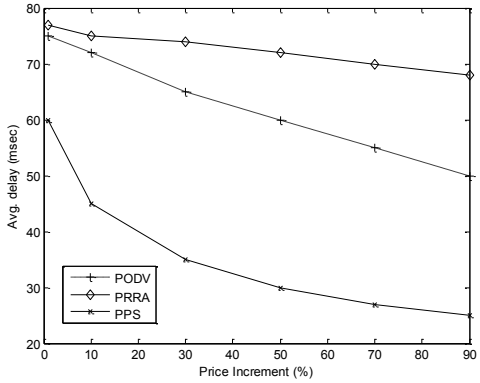

FIGURE 17. Average delay vs. Price increment percentage (100 GWs, 5-30 DCs, 4500 LNs).

Fig. 17 shows that as the GWs price increase,PPS provides delays that are much lower than PODV and PRRA do. This is because PPS's resource management algorithm is based on this price consideration. These results indicate that our pricing scheme is successful in providing lower delivery delay given the incentive to do so (i.e. higher data price). We note that PODV performs better than PRRA in this regard since PODV's routing is essentially based on reducing hop-count on end-to-end links, indicating better handling of end-to-end delay on the LN-to-GW level.

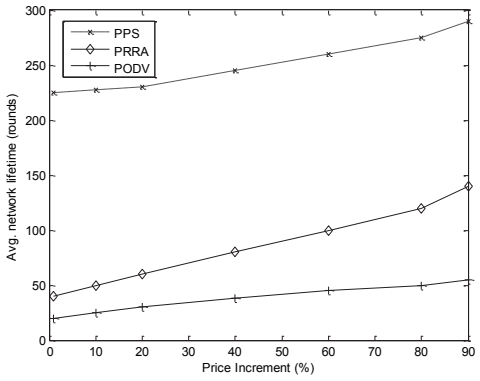

FIGURE 18. Average network lifetime vs. Price increment (100 GWs, 5-30 DCs, 4500 LNs).

Fig. 18 shows how the three delivery schemes react to price increments with respect to ANL per peripheral network. Here,

as the client pays higher prices, more LNs (represented by GWs interfacing each peripheral network) will participate in the public sensing service, which consequently entails more consumption of their energy resources. A higher price would then diversify the sources of data giving away to a longer network lifetime. Again, PPS shows that it successfully utilizes its pricing function in a manner that conserves lifetime and provides it as a resource for higher bidders only. Here, we note that PRRA performs better since its algorithm does not overwhelm every node in the network by periodical table updates that consequently dramatically degrade the nodes' battery power. This is the case with PODV which performs worst in terms of lifetime regardless of the steady increments on GW pricing.

VI. CONCLUSION

In this paper we introduce PPS - a priced IoT public sensing framework for smart cities. Our framework is based on a multi-tier architecture that caters for heterogeneous data sources (e.g. sensors) in addition to stationary and dynamic data collectors in peripheral networks that are assumed to be connected to a data cloud via more powerful intermediary gateways. According to our framework, access points at the top of the architecture receive user queries and initiate data requests. Our delivery scheme implements algorithms that realize delay-sensitivity and user-based data quality requirements. Moreover, we provide a dynamic two-tier pricing model that acts on both ends of the supply-demand chain. At the lower tier, it caters for the social welfare of the peripheral system by observing constraints on delay, GW capacity and system lifetime. At the top tier, our pricing model employs a utility function that maximizes the client's gain according to the delay limit, service quality, trust factor and the monetary value of the requested data. We provide simulation results showing the efficiency of our framework when compared to two prominent mobile Ad-hoc data delivery protocols. Our simulation results show that the PPS framework exhibits superior performance for different network sizes, lifetime, end-to-end delays, data prices and packet delivery ratios.

Future work would investigate utilizing vehicles in smart city settings either as sensors/reporters or mobile data collectors with semi-deterministic mobility trajectories. It is also interesting to look at the application of localization methods among sensors operating on different technologies and to study the effect of such methods on the system's performance and the delivery rate.

REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: A wireless and mobility-related view," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 44–51, Dec. 2010.

[3] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The rise of people-centric sensing," *IEEE Int. Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.

[4] D. Philipp, F. Durr, and K. Rothermel, "A sensor network abstraction for flexible public sensing systems," in *Proc. IEEE Int. Conf. MASS*, Oct. 2011, pp. 460–469.

[5] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: Towards cooperation frameworks for open innovation," in *The Future Internet* (Lecture Notes in Computer Science). New York, NY, USA: Springer-Verlag, 2011, pp. 431–446.

[6] D. Zhang, B. Guo, B. Li, and Z. Yu, "Extracting social and community intelligence from digital footprints: An emerging research area," in *Proc. 7th Int. Conf. UIC*, Oct. 2010, pp. 4–18.

[7] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels, "Citizen noise pollution monitoring," in *Proc. 10th Annu. Int. Conf. Digit. Government Res.*, May 2009, pp. 96–103.

[8] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "BubbleSensing: Binding sensing tasks to the physical world," *Pervas. Mobile Comput.*, vol. 6, no. 1, pp. 58–71, 2009.

[9] E. Kanjo, S. Benford, M. Paxton, A. Chamberlain, D. S. Fraser, D. Woodgate, D. Crellin, and A. Woolard, "MobGeoSen: Facilitating personal geosensor data collection and visualization using mobile phones," *Pers. Ubiquitous Comput.*, vol. 12, no. 8, pp. 599–607, 2008.

[10] P. Baier, H. Weinschrott, F. Dürr, and K. Rothermel, "MapCorrect: Automatic correction and validation of road maps using public sensing," in *Proc. 36th Ann. IEEE Conf. LCN*, Oct. 2011, pp. 58–66.

[11] A. T. Campbell, S. B. Eisenman, N. Lane, E. Miluzzo, and R. Peterson, "People-centric urban sensing," in *Proc. 2nd Annu. Int. Workshop Wireless Int.*, 2006, pp. 18-31.

[12] H. Patni, C. Henson, and A. Sheth, "Linked sensor data," in *Proc. Int. Symp. CTS*, 2010, pp. 362–370.

[13] R. Golchay, F. L. Mouel, S. Frénot, and J. Ponge, *Towards Bridging IoT and Cloud Services: Proposing Smartphones as Mobile and Autonomic Service Gateways*. Ithaca, NY, USA: Cornell Univ. Press, Jul. 2011.

[14] F. Hao, T. V. Lakshman, S. Mukherjee, and H. Song, "Enhancing dynamic cloud-based services using network virtualization," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 67–74, 2010.

[15] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter cities and their innovation challenges," *IEEE Comput.*, vol. 44, no. 6, pp. 32–39, Jun. 2011.

[16] J. Lee, S. Baik, and C. Lee, "Building an integrated service management platform for ubiquitous cities," *IEEE Comput.*, vol. 44, no. 6, pp. 56–63, Jun. 2011.

[17] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proc. ICECC*, 2011, pp. 1028–1031.

[18] C. Perkins, E. Belding-Royer, and S. Das, "Ad-hoc on-demand distance vector (AODV) routing," in *Proc. IETF Netw. Work. Group RFC*, Jul. 2003, pp. 90–100.

[19] D. Johnson and D. Maltz, "Dynamic source routing in ad-hoc wireless networks," *Mobile Comput.*, vol. 353, no. 5, pp. 153–181, Aug. 1996.

[20] C. A. Gizelis and D. D. Vergados, "A survey of pricing schemes in wireless networks," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 1, pp. 126–145, Jan./Mar. 2011.

[21] G. V. Ozianyi, N. Ventura, and E. Golovins, "A novel pricing approach to support QoS in 3G networks," *Comput. Netw.*, vol. 52, no. 7, pp. 1433–1450, May 2008.

[22] J. W. Lee, R. R. Mazumdar, and B. N. Shroff, "Joint resource allocation and base-station assignment for the downlink in CDMA networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 1–14, Feb. 2006.

[23] K. R. Lam, D. M. Chiu, and C. Lui, "On the access pricing and network scaling issues of wireless mesh networks," *IEEE Trans. Comput.*, vol. 56, no. 11, pp. 1456–1469, Nov. 2007.

[24] L. Cheng, C. Chen, J. Ma, and Y. Chen, "A group-level incentive scheme for data collection in wireless sensor networks," in *Proc. 6th Annu. IEEE CCNC*, Jan. 2009, pp. 1–5.

[25] *Family of Standards for Wireless Access in Vehicular Environments (WAVE)*, IEEE Standard 1609, Jan. 2006.

[26] L. Buttyan and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proc. 1st ACM Int. Symp. MobiHoc*, 2000, pp. 87–96.

[27] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modeling incentives for collaboration in mobile ad-hoc networks," *Ad-Hoc Wireless Netw.*, vol. 57 no. 4, pp. 427–439, Aug. 2004.

[28] K. C. Lan, C. M. Chou, and H. Y. Wang, "An incentive-based framework for vehicle-based mobile sensing," in *Proc. 2nd Int. Workshop Sensor Netw. Intell. Gathering Monitoring*, Aug. 2012, pp. 1152–1157.
[29] J. Bohli, C. Sorge, and D. Westhoff, "Initial observations on economics, pricing, and penetration of the internet of things market," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 2, pp. 50–55, Apr. 2009.
[30] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Intl. Conf. Syst. Sci.*, Jan. 2000, pp. 1–10.
[31] W. Alsalih, H. Hassanein, and S. Akl, "Routing to a mobile data collector on a predefined trajectory," in *Proc. IEEE ICC*, Jun. 2009, pp. 1–5.
[32] J. Li and Y. Q. Song, "DLB: A novel real-time QoS control mechanism for multimedia transmission," *Int. J. High Perform. Comput. Netw.*, vol. 6, no. 1, pp. 4–14, 2009.
[33] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 2, pp. 265–278, Apr./Jun. 2012.
[34] J. Carbo, J. M. Molina, and J. Davila, "Trust management through fuzzy reputation," *Int. J. Cooperat. Inf. Syst.*, vol. 12, no. 1, pp. 135–155, 2003.
[35] M. H. Mamoun, "A new reliable routing algorithm for MANET," *Int. J. Res. Rev. Comput. Sci.*, vol. 2, no. 3, pp. 638–642, Jun. 2011.

ASHRAF E. AL-FAGIH (M'99) is an Assistant Professor with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He received the B.Sc. degree in computer engineering at KFUPM in 2000, the master's degree in computer science from Texas A&M University, College Station, TX, USA, in 2004, and the Ph.D. degree in computing from Queen's University, Kingston, ON, Canada, in 2013. His current research interests include wireless ad-hoc and sensor networking, delay-tolerant networking, information-centric networks, RFID systems, and the realization of the Internet of Things. Technical papers he peer reviewed were presented in IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the *IEEE Communications Surveys and Tutorials*, the *Journal of Network and Computer Applications*, IEEE GLOBECOM, IEEE ICC, and IEEE LCN.

FADI M. AL-TURJMAN (M'07) is an Assistant Professor with the School of Engineering, University of Guelph, Guelph, ON, Canada. He is working in the area of wireless networks architectures, deployments, and performance evaluation. He received the Ph.D. degree in computer science from Queen's University, Kingston, ON, in 2011, and the B.Sc. (Hons.) and M.Sc. (Hons.) degrees in computer engineering from Kuwait University, Kuwait City, Kuwait, in 2004 and 2007, respectively. He has authored or co-authored more than 40 reputable journal and international conference papers, in addition to chairing a number of workshops in international symposia and conferences; including the FTRA IET in MUSIC in 2012, the IEEE WLN in LCN in 2012 and 2013, and the IEEE G-IoT in GLOBECOM in 2012.

WALEED M. ALSALIH is an Assistant Professor with the Computer Science Department, King Saud University, Riyadh, Saudi Arabia. He received the M.Sc. and Ph.D. degrees from the School of Computing, Queen's University, Kingston, ON, Canada, in 2005 and 2009, respectively. He has served in the program committee of some international conferences and workshops, and he is currently an Editorial Board Member of the *Parallel Processing Letters*. His current research interests include wireless sensor networks, RFID, mobile and pervasive computing, and discrete optimization.

HOSSAM S. HASSANEIN (M'86–SM'01) is a leading researcher in the areas of wireless and mobile networks architecture, protocols, and services. He has published more than 500 publications in journals, conferences and book chapters, in addition to numerous keynotes and plenary talks in flagship venues. He has received much recognition and several Best Paper Awards at top international conferences. He is the Founder and Director of the Telecommunications Research Laboratory, Queen's University School of Computing, Kingston, ON, Canada, with extensive international academic and industrial collaborations. He is a Former Chair of the IEEE Communication Society Technical Committee on Ad-hoc and Sensor Networks. He was the IEEE Communications Society Distinguished Speaker (Distinguished Lecturer from 2008 to 2010.