An Energy-Efficient Service Discovery Protocol for the IoT based on a Multi-Tier WSN Architecture

Rana Helal
German University in Cairo
Department of Media Engineering and Technology
Cairo, Egypt
Email: rana.abdelshafi@guc.edu.eg

Amr ElMougy
German University in Cairo
Department of Media Engineering and Technology
Cairo, Egypt
Email: amr.elmougy@guc.edu.eg

***Abstract*—The Internet of Things is expected to foster billions of heterogeneous sensors and actuators that support a wide range of applications. Maximizing the utilization of these sensors is hinged upon the ability to discover their capabilities (ex: measured attributes, location, accuracy, etc.). This process is known as Service Discovery (SD), and can have a local scope (SD within a bounded area), or a global scope. Traditionally, SD is performed using a dedicated gateway that is always active and connected to the Internet. This gateway stores the attributes of sensors within its area. However, this approach may be too expensive in the presence of billions of sensors. In this paper, a novel protocol for local SD is proposed that eliminates the need for a dedicated gateway. The protocol utilizes a multi-tier network architecture in order to achieve two main objectives: energy efficiency and high success rate in satisfying service requests. Energy efficiency is achieved by limiting the number of hops that an SD request has to traverse before being satisfied, while a high success rate is guaranteed using a hierarchical structure of Distributed Hash Tables (DHTs), where information regarding sensor capabilities is stored. Extensive computer simulations are used to evaluate the performance of the proposed protocol in comparison to single level architectures and gateway-based solutions. The results show that the proposed protocol achieves energy efficiency without sacrificing success rate of serving requests.**
***Keywords*—-Internet of Things(IoT), Service Discovery (SD), Constrained Application Protocol (CoAP), ontology, Distributed Hash Tables (DHT), Wireless Sensor Network (WSN), hierarchical architecture**

I. Introduction

The Internet of Things (IoT) is no longer a vision for the future. Every day more and more smart objects are being connected to the Internet, enabling new sophisticated and exciting applications such as smart homes, interactive pollution maps, real-time traffic information, and many more. Thus, the vision of a smart city [1] [2], where ubiquitous information is available from objects all around us, is now being realized.

The growth of the IoT is leveraged mainly by recent advancements in sensor technology, which offer scalable and inexpensive solutions that were not possible before. Companies are now marketing programmable sensor boards that have small size, powerful processors, HD video codecs, memories up to 5GB, and can support several communication technologies such as WiFi, Bluetooth Low Energy (BLE), and ZigBee. In addition, a very large number of inexpensive sensing devices are now available for measuring a wide range of attributes such as temperature, light, carbon monoxide, and many more. For these reasons, it is predicted that billions of sensors will be connected to the Internet by 2020 [3].

In order to maximize the utilization of sensors and minimize costs, these sensors will have to remain operational for extended periods of time (months or even years). They also need to self-configure in order to limit human intervention. Moreover, the development of advanced IoT applications hinges upon the ability to discover the capabilities of the sensors. For example, when a smartphone enters into a region, how can it discover the types of sensors available, their locations, accuracy, etc.? This process is known as Service Discovery (SD). It can be done locally, where the objective is to discover sensors within the smartphones region (as in the above example), or globally, where SD is performed for remote sensors.

It is clear that semantics are highly important in SD. For example, a researcher studying pollution in a certain region should be able to search for sensors in particular GPS coordinates, with carbon monoxide sensors, and sufficient battery power to support an extended study [4] [5]. Services supported by the sensors are described using strings of characters known as Uniform Resource Identifiers (URIs) [6]. Thus, a URI can be used to describe sensor locations, accuracy, etc. The objective of SD becomes to learn the URIs of sensors. This can be highly challenging for many reasons. First, the large number of sensors makes data mining difficult. Second, the topology is dynamic, with nodes dying and new ones joining constantly. Third, self-configuration is a critical requirement, especially for Machine-to-Machine (M2M) communications.

Traditional solutions to SD generally require the presence of a gateway [7] [8], which holds sensor metadata within a small area and is always active and connected to the Internet. However, if billions of sensors are to be implemented, then billions of gateways may possibly be required, which will probably cost much more compared to regular sensors. In addition, gateways represent a single point of failure for the sensors they manage. Thus, it is desirable to have a solution that is complementary to gateways.

This paper presents a novel solution for SD that eliminates the need for a persistent gateway. The proposed solution focuses on local SD, while global SD remains part of our future work. Thus, the problem tackled in this paper can be formulated as follows: how can a smartphone automatically discover the URIs of sensors, not only within its direct communication range, but within its extended neighborhood, without the presence of dedicated gateways.

To tackle this problem, our solution proposes a multi-tier Wireless Sensor Network (WSN) hierarchy that targets the maximization of energy efficiency and success rate of satisfying requests. Note that a request will be sent by smartphones containing a list of required sensor measurements with particular attributes. A request is considered fully satisfied if sensors are found for all required measurements with the specified attributes, and partially satisfied if sensors are found for only a part of the required attributes. In the proposed hierarchy, periodically elected Cluster Heads (CHs) store URIs of sensors within their 1-hop neighborhoods and satisfy SD requests on behalf of these sensors. If a CH fails, another one can simply be elected. Requests that cannot be satisfied at the CHs are relayed to a higher level of the hierarchy called area routers. These area routers aggregate the services offered at the CHs in their areas and can communicate with each other to extend the boundaries of the SD region. Thus, energy efficiency is achieved by limiting the number of hops that a request has to traverse before being satisfied, while a high success rate is guaranteed since requests that end up at the area routers will definitely be answered. In addition, the proposed protocol utilizes dynamic sleep scheduling according to the population density of smartphones within the CHs region. This leads to further improvements in energy efficiency.

The remaining sections of this paper are organized as follows: Section II reviews some recent related work pertaining to our research, Section III discusses the full details of our proposal, Section IV includes experimental evaluation of the protocol, while Section V provides some concluding remarks.

II. Related Work

In the past decade, WSN had become of great importance in a vast variety of fields such as environment monitoring, healthcare and several industrial applications. Since sensors have limited capabilities as pointed to earlier, communication between these sensors need to be energy efficient to avoid battery depletion. For this reason, most sensors either use ZigBee or BLE as the main networking standards. BLE is the new version of the standard Bluetooth protocol having a smaller communication range (10m) but consuming much lower energy. In addition, the proliferation of BLE on smartphones makes it very suitable for end devices and applications requiring human interactions. The main disadvantage of BLE is that it does not support multi-hop communication. For large WSNs, this will impose a significant limitation since serving user requests will rely on propagating messages along several hops. ZigBee, on the other hand, can solve this issue since it supports multi-hopping without drastically compromising energy consumption. Nowadays, some sensor modules provide support for both technologies on one chip to make use of the best of both worlds.

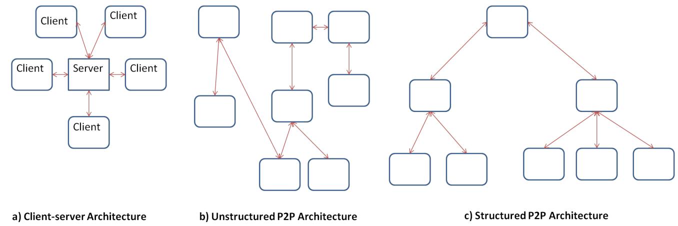

Fig. 1. The three possible network architectures

To support SD, most WSNs in literature have a hierarchical architecture [9]. The lowest level of the hierarchy will be the sensors providing several readings. Usually, several gateways are deployed across the network to be responsible for a subset of sensor nodes. In [10], the authors assume that each group of sensors form a Network of Things (NoT) that is isolated from the other NoTs and can only communicate with users using a gateway. This gateway is responsible for checking the user request against the services provided by its NoT and can combine multiple data values to provide better response for the request. A similar approach was used in [8] where each WSN has a dedicated gateway that works in one of two modes. The first mode is acting as a translator from user HTTP requests to Constrained Application Protocol (CoAP) [11] (CoAP is discussed later in this section) requests while the second one is allowing sensors to publish their data on this gateway so that users can access it later.

Despite the fact that these papers achieve SD, they depend on a single entity to act as an intermediary between users and sensor networks. This can be quite problematic due to the fact that if this gateway fails, the whole network connected to it will not be accessible. For this reason, several papers focusing on SD have been trying to find other ways eliminating the need for a gateway. For example, distributed consensus decision-making was used in [12] in order to enhance SD in the IoT. This was achieved by modelling the IoT network as a hierarchical graph of vertices (sensor nodes) where each vertex produces a local decision regarding the composition of available services, and then an iterative algorithm is used to reach a global decision related to the requested service. In [13], knowledge bases were distributed among several locations and linked together to provide better relations discovery. In these papers and others, sensors need to send information back and forth to each other in a Peer-to-Peer (P2P) architecture.

In comparison to client-server architecture, P2P provides a better structure in terms of scalability. P2P networks could be of two types; structured and unstructured [8] [13] as shown in Figure 1. If SD is going to be applied in an unstructured network, any user request should be broadcasted to all sensors in the network which is not energy-efficient. For this reason, structured P2P architecture such as Distributed Hash Tables (DHT), which is used in this paper, are preferable in the field of IoT. DHTs can store translations of sensor URIs, providing an efficient approach for SD.

Hash tables in general map each value to a unique key. A node that needs to access a certain value will search for it using its key. In DHT, the hash table key-value pairs are distributed across the network nodes eliminating the single point of failure problem associated with gateways. DHT provides a scalable solution to the massive number of nodes that are known for their dynamic nature. Although DHT-based architectures are similar to Domain Name Systems (DNS), there are a lot of core differences in their mode of operation [8]. DNS can not be used with IoT since it requires full domain names to be resolved to an IP address and can not work with URIs.

Finally, it is worth mentioning that currently the most promising protocol for exchanging messages between sensors and their applications is CoAP [14] [15], which is an application layer protocol developed primarily for constrained devices. CoAP is basically a lightweight version of HTTP and is also based on client/server communications. However, it uses UDP at the transport layer which makes it more energy efficient. Since CoAP is compatible with HTTP, high-end nodes can use HTTP to connect to the Internet while the low-level network nodes like the sensors can communicate using CoAP and translation can be done easily between the two formats. In this protocol, there are two types of messages, request and response, both with a small header size that is suitable to the nature of sensors in a WSN.In our proposed protocol, CoAP will be used at the application layer.

III. Service Discovery Protocol

As mentioned before, the SD protocol proposed in this paper utilizes a multi-tier architecture to guarantee high success rate while maintaining energy efficiency. The multi-tier architecture contains three types of nodes: area routers, CHs, and regular sensor nodes. We assume that the geographical region to be covered is divided into areas, where each area has one area router. The area routers in different areas form a 1-hop network. In addition, each area router manages all CHs in its area. Each CH stores the attributes of a few regular sensors within its direct communication range, while area routers store the attributes that can be found at each CH in its area. This architecture is illustrated in Figure 2 for a region of 4 areas.

Attributes at the CHs and area routers are stored as DHTs, where each hash table stores pairs of URIs and the services supported by each sensor. Upon joining the network, the sensor will listen to beacon messages periodically broadcasted by the CH. It will continue to listen long enough to give a chance if a sleeping nearby CH is available to wake up and send its beacons. If not found, the sensor node will elect itself as CH. In the case that this CH is forming a single-node cluster and there is another nearby CH that was discovered later, the two clusters can be joined together. However, if a CH is found from the beginning, the sensor will publish to it the services it provides. For example if sensor with URI value sensor1 provides readings for temperature and pressure, an entry will be added in the hashtable of its corresponding CH as *<sensor1,temperature;pressure >*. If the CH is capable of adding this sensor to its hash table, it will respond with a positive reply. Otherwise, the sensor will have to continue

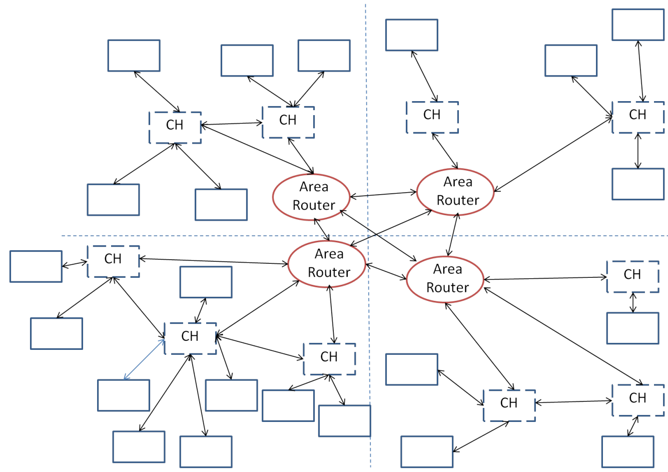

Fig. 2. Example of a network divided into areas

searching for another CH or elect itself as a CH. Upon being elected as a CH, a node will have to send beacon messages periodically to announce its presence. If a CH fails, its sensors will stop receiving beacon messages and will have to restart the CH discovery process.

Each CH has to register itself with its area router by aggregating the attributes it serves and sending them to the router. If the services in the table change, the CH informs the area router. This way, routers will know all attributes found in their regions. Note that CHs and area routers can be regular sensor nodes that exchange their role with other nodes in their neighborhood to avoid rapid battery depletion. As mentioned before, current sensor technology supports relatively large memories and several communication technologies capable of satisfying the above requirements. Thus, the proposed architecture eliminates the single point of failure associated with dedicated gateways, while guaranteeing high success rate for an extended area. The following sections illustrate how CHs and routers process SD requests.

A. CH Role

The main task of the CH is to receive and process SD requests from smartphones. When a smartphone comes within communication range of a CH, it will start hearing its beacon messages. At this point, the smartphone can start sending it CoAP requests. This request consists of the URI of the smartphone and the services needed by the application. For example, a weather application may send a request for temperature, humidity and wind speed measurements. If this request came from smartphone with unique identifier of value 1, the request will be formatted as *"request-temperature,humidity,wind speed;r01"*.

When the CH receives a request, it checks its hashtable for sensors that satisfy this request. If the request can be fully satisfied, the CH will formulate a reply to the smartphone. In the case where several sensors can provide the same requested service, the reply will contain the URIs of both sensors.

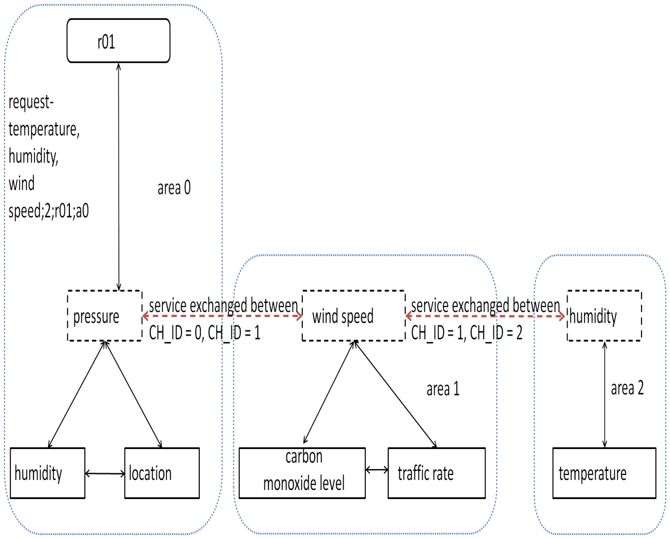

Fig. 3. Services exchanged between CHs

However, if the request cannot be fully satisfied, it will be propagated to the 1-hop neighboring CHs in order to increase the hit rate. Naturally, more hops can be allowed but this will come at the expense of energy consumption. To support this 1-hop exchange, each CH will store entries in it hash table about its neighboring CHs. This entry will have a key value corresponding to the CH_ID and a value consisting of the URIs of the sensors (with their attributes). For instance, the network in Figure 3 consists of 3 areas each with its CH. Each rectangle represents a sensor with the service it provides and dashed ones are the CHs for every area (note that CHs may also provide services, as shown). Initially, CH_ID = 0 and CH_ID = 1 exchange their sensor services, as well as CH_ID = 1 and CH_ID = 2. For example, the exchange between CH_ID = 0 and CH_ID = 1 results in an entry with key *"CH_ID = 1"* and value *"sensor3:wind speed,sensor4:carbon monoxide level,sensor5:traffic rate"* to be added to the hash table of CH_ID = 0. To illustrate how a CH processes a SD request, consider the message exchanges shown in Fig. 4. Here, the process starts with sensors (only Sensor1 is shown for clarification) associating themselves with the CH (CH0 in this case). Then, CH1 sends its services to CH0. Afterwards, when CH0 receives an SD request from smartphone, r01, for temperature, humidity, and wind speed measurements, as shown in the figure, CH_ID = 0 will reply to the smartphone *r01* that sensor 1 can provide a reading for humidity and sensor 3 can provide a reading for wind speed. The service temperature, however, can only be provided at CH_ID = 2 (as shown in Fig. 3), which is unreachable from CH_ID = 0.Thus, CH_ID = 0 will not be able to satisfy this part of the SD request. The number of CH_IDs to be saved on each CH depends on the storage capacity. Since the CH is a normal sensor, it is not going to be capable of storing all nearby areas data. However, even if a CH stores a small number of neighboring areas, this will help in increasing the hit rate.

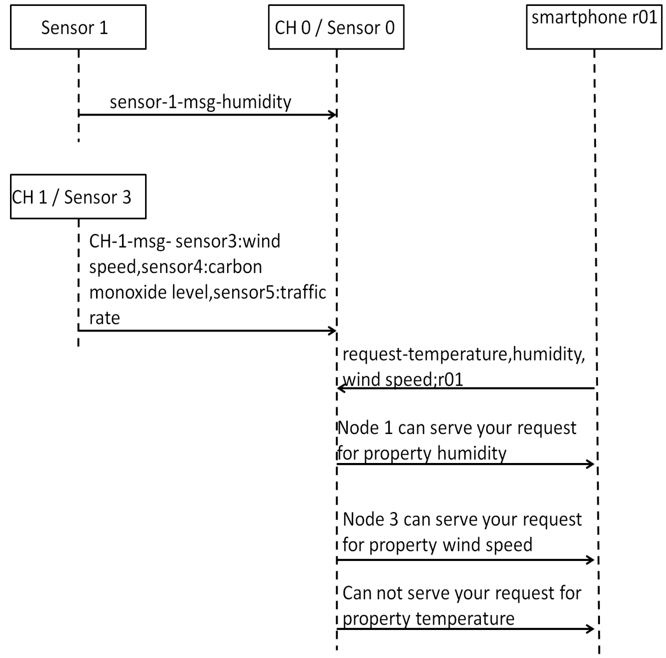

Fig. 4. Example of exchanged messages over time

Till this point, even after enabling direct communication between CHs, we can not guarantee 100 % hit rate. Thus, to achieve this success rate, requests that cannot be satisfied at this level will be sent to the area router.

B. Area Routers Role

The purpose of area routers is to extend the SD region beyond the ranges of a CH. Each area router maintains a hash table that stores the services that can be satisfied at each CH. Since the area router may potentially serve a large number of sensors, it may be flooded with a large number of requests, depleting its battery. Thus, the request initially sent by the smartphone will contain a flag to specify if the user wants the request to be sent to the area router (if needed) or not. For example, in our weather application, the request will change to *"request-temperature,humidity,wind speed;1;r01;1"*. Thus, non-critical SD requests may opt to risk incomplete replies to save the batteries of the area router.

Figure 5 shows an example for three areas with the addition of area routers; area 0 has its own area router while areas 1 and 2 belong to the same area router. Assume that the above request was sent from the smartphone *r01* to the CH in area 0. Figure 6 shows the sequence of exchanged messages in case that area routers are enabled. Since *"humidity"* can be provided by the same area, the CH will reply directly to the smartphone. On the other hand, the two other services are not available in this CH so they are sent to *area router 0*. This area router also does not find any other CH in its area that can satisfy this request so it forwards the request to *area router 1*, which finds the two services in the two areas it connects to. *Area router 1* will then reply to *area router 0* with the URI of the sensor that can provide this service which replies to the

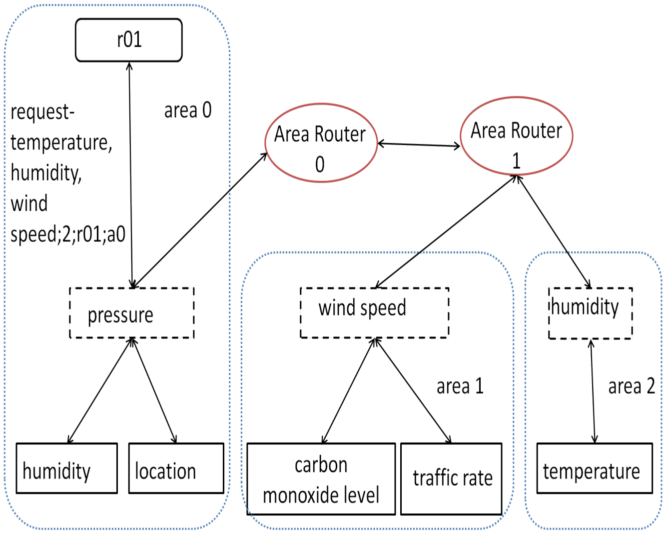

Fig. 5. Sample network with addition of area routers

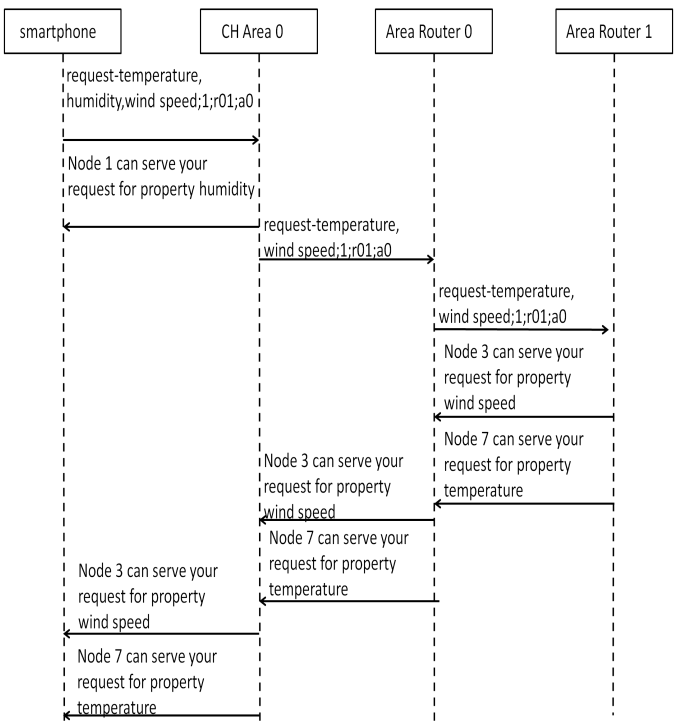

Fig. 6. Example of exchanged messages over time in case of enabling area routers

CH in area 0, replying to the smartphone. In order to obtain readings from these sensors, the same path can be followed through area routers back to the smartphone.

In order to summarize the proposed protocol, Figure 7 shows a flowchart that gives an overview of how SD is achieved if area routers are enabled.

C. Dynamic duty cycling based on smartphone density

In order to ensure energy efficiency, the proposed protocol incorporates dynamic duty cycling based on the density of smartphones with the region of each CH. Every sensor has a duty cycle that varies between two states; awake and sleeping. In the first state, the sensor consumes more energy and is active. On the other hand, in the sleeping state, the sensor is dormant but consumes a very small amount of energy. In this state, the sensor can not transmit or receive any messages.

The traditional implementation of duty cycles is defining the duration in which the sensor is in the active state. Our implementation suggests that as the number of smartphones in an area increases, the number of user requests will increase, therefore the number of cycles in which the sensor node is active should increase as well. However, if the number of smartphones is small, it will be better for the sensors to sleep for a longer time, thus conserving energy. Thus, every CH will calculate its preferred duty cycle based on monitoring the transmissions of the smartphones within its region. The CH will then notify its neighbors of this duty cycle since they do not have a method to detect smartphone density on their own. Several experiments were carried out in order to find the optimal percentage to be used with a sample network similar to the one that was shown in Figure 5 and they are explained in details in the following section.

IV. Experimental Results

After implementing the whole framework, several experiments were conducted in order to evaluate its performance. The main factors that illustrate the efficiency of the proposed protocol are the hit rate (the percentage of requests that can be served compared to their total) and the energy consumption of sensors, CHs and area routers. Network simulations were built using OMNet++. The network used consisted of multiple clusters, each containing its own CH and a number of sensors varying between 1 and 5. Randomly placed mobile nodes were added to represent the smartphones issuing requests. Each mobile node updates its location at random time intervals and connects to the nearest CH. Furthermore, several area routers were used according to the number of areas in the experiment. Each area router was responsible for 4 adjacent CHs. A snapshot of part of the network described can be seen in Figure 8. Five clusters consisting of a CH (oval nodes) and 2 sensors (rectangles) are shown, along with 2 area routers (rectangles with a star) and a number of smartphones (rectangles with a wheel) scattered in these clusters. The CHs of the 4 clusters belonging to the same area router can have direct communications only with their 1-hop neighbors. For instance, *sensor0* can exchange messages with *sensor3* but can not communicate with the other CHs except through *areaRouter0*. In the coming sections, the results for the experiments conducted are presented.

A. Multi-tier Architecture Evaluation

The setup of the network in this experiment consists of 16 different clusters, each with its own CH and 2 other sensors. 4 area routers were needed since each area router was responsible for 4 clusters. Additionally, a variable number of smartphones were present in arbitrary areas to issue multiple

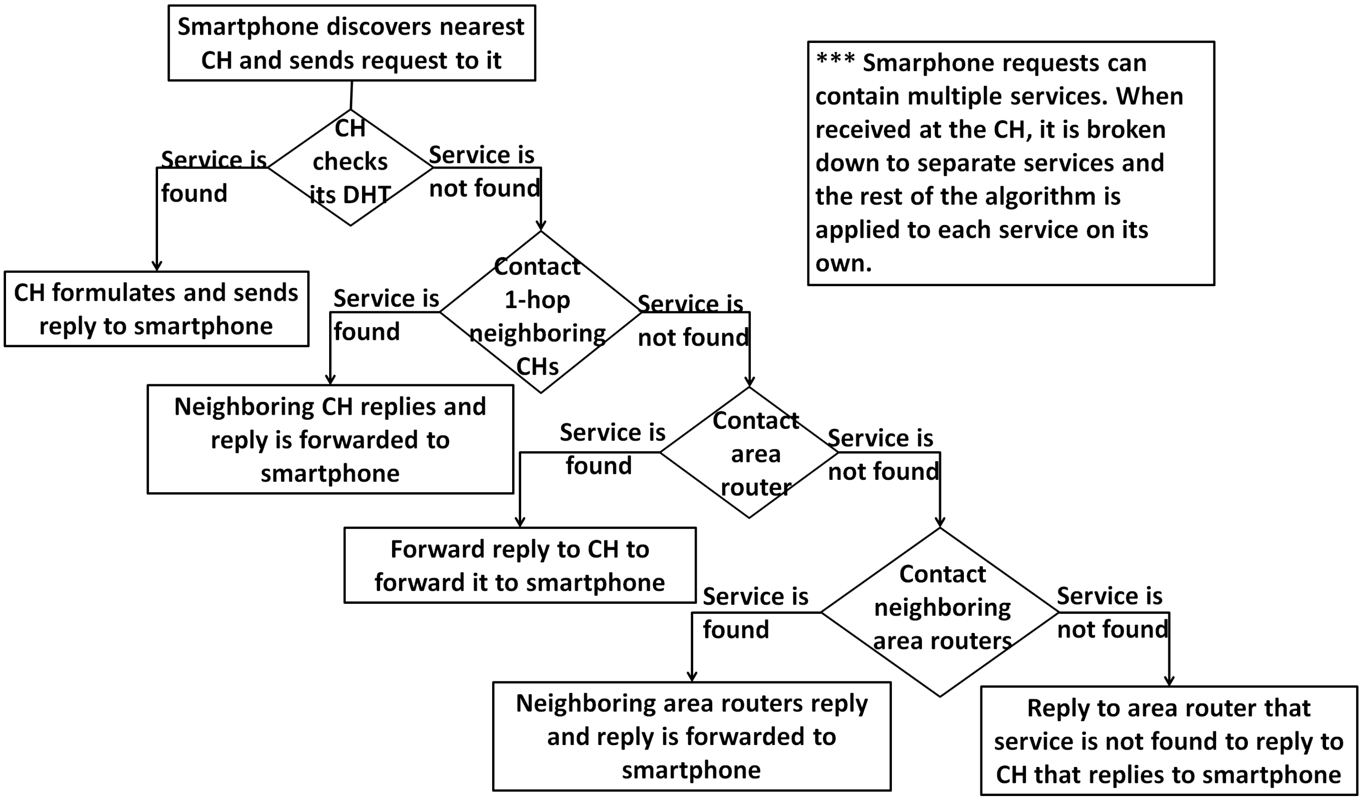

Fig. 7. Flowchart of the proposed protocol

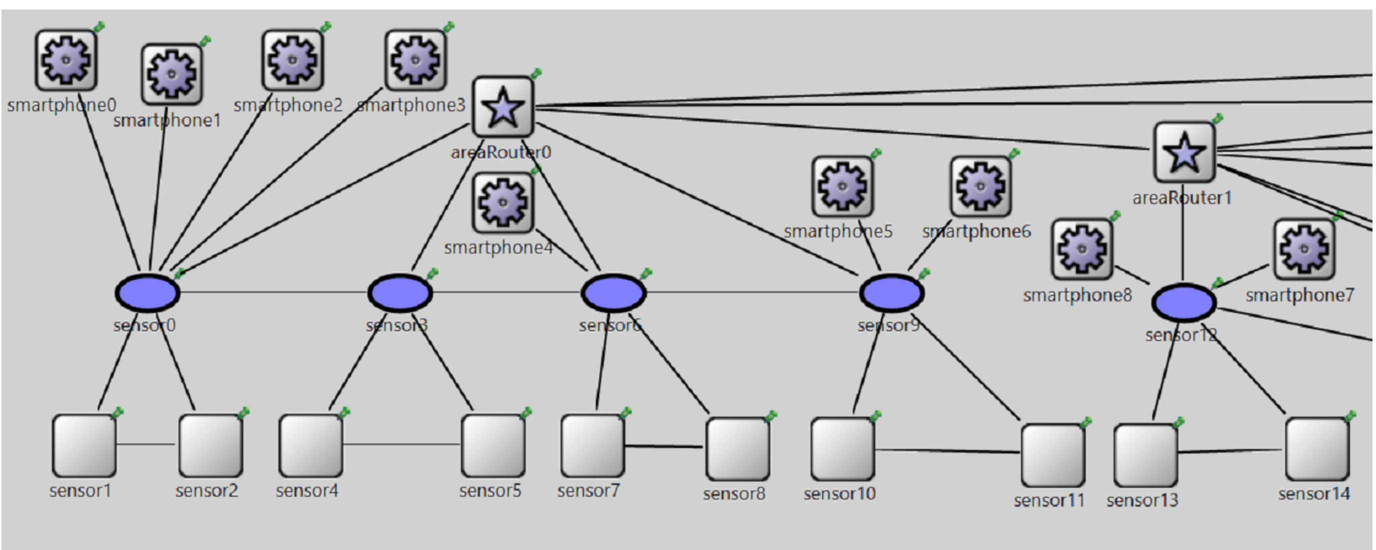

Fig. 8. Part of simulation network used in experiments

requests. The requests were generated randomly from a set containing 20 different services such as temperature, humidity, pressure, acidity level and so on. Each smartphone sent a request with 2 different services at a time and repeated this with a different combination for 3 times at random time intervals making the total number of requests reach $6n$ where n is the number of smartphones. In this experiment, a fixed duty cycle was chosen with a total period of 30 time units (20 units awake and 10 units sleeping). Three different situations were tested; 1- using area routers and direct communications between CHs, 2- No area routers but with direct communications between CHs, and 3- No area routers nor direct communications between CHs. The results for hit rate as well as energy consumption of the CHs and area routers involved

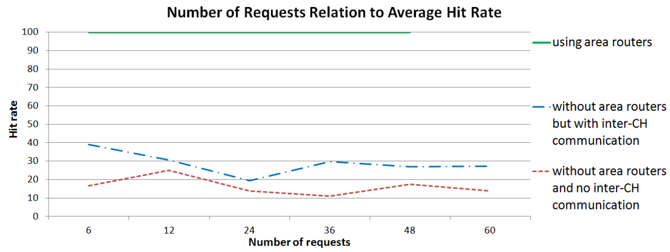

Fig. 9. Requests number in relation to hit rate

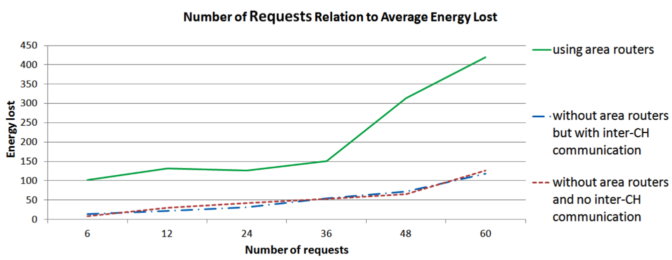

Fig. 10. Requests number in relation to energy lost

are plotted in Figure 9 and Figure 10.

1) Hit Rate: As expected, the hit rate is always 100 % when area routers were used. This is due to the fact that all area routers can communicate together in a single hop fashion to satify requests. From Figure 9, if area routers are not enabled but, direct communication between adjacent CHs is enabled, the hit rate drops to a value between 20 and 40 %. This is due to the fact that in this experiment, only one nearby area can be accessed. When disabling both previous multi-hopping modes, the hit rate is much lower reaching a percentage between 10 and 25 %. This proves that multi-hopping whether through area routers or through CHs will serve more user requests.

2) Energy Consumption: The increase in hit rate shown in the previous section is motivating to always enable area routers. However, doing this will increase the energy consumed from the sensors. The rate of energy consumed is calculated in milliWatt's/second (mW/sec) through the amount deducted from power over time. The initial energy is calculated as $current/hour \times 60 \times 60 \times voltage$. This equation was implemented in a pre-defined battery model in OMNet++ [16]. The numbers for the current/hour and voltage were determined according to real sensors in the market [17]. For instance, the current was 0.5 mA and voltage was 4 V leading to an initial energy value of 7200. It is clear from Figure 10 that using area routers consumed the highest energy value. Multi-hopping between CHs and without multi-hopping at all have similar values with small number of requests. As the number of requests increase, multi-hopping will be needed. Thus, direct communication between multi-hops shows more energy consumption.

From this we can conclude that area routers should be involved with the critical user requests, while the other requests can be served using multi-hopping through CHs if needed in order to achieve the best hit rate and in the same time conserve

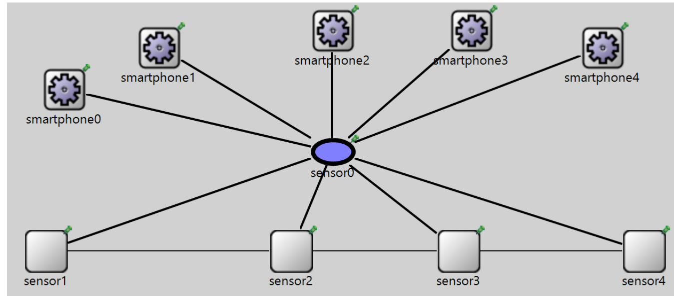

Fig. 11. Simulation network used in experiments regarding dynamic duty cycling

energy as much as possible.

B. Dynamic duty cycling based on smartphone density

In order to vary the number of wake up cycles as a function of smartphones' number, we multiplied the smartphones' number by a certain percentage. This experiment was repeated with three different equations where the number of wake up cycles is $20\%, 50\%$ and $80\% \times smartphonesnumber$. In order to benchmark the effect of varying the cycles in one period, two more experiments were carried out whith constant duty cycles-the first experiment assumed a 50% duty cycle, while the second one used 100% duty cycle. The total period is 30 time units. The smartphones' number varied from 1 per area to 30 per area and all the requests had a hop count of 0 (disabling area routers and multi-hopping through CHs). Each request has 2 services out of 10 possible services. A screenshot of the network used in the coming experiments is shown in Figure 11. Since all forms of multi-hopping were disabled, one cluster only is tested. This cluster had a total of 4 sensors along with the CH. The hit rate and energy consumption of the CH were measured and the results are discussed below.

1) Hit Rate: As we can see in the graph in Figure 12, when we increase the percentage of wake up cycles, the hit rate increases reaching a maximum of 53 % in the case of 80 % of the smartphones' number. Fixed duty cycling has a smaller range of variance since they are not related to the smartphone's number. The 50 % duty cycle gives a hit rate between 30 and 40 % and the 100 % duty cycle achieves a hit rate ranging from 45 till 55 %. Please note that the maximum hit rate is about 50 % due to the fact that every area has 5 sensors only (providing 5 services) while the total possible services in a request were 10 which is double the number of services provided per cluster.

2) Energy Consumption: In this experiment, calculations regarding the initial energy values and energy loss are the same as experiments in section IV-A2. The graph in Figure 13 is very similar to the hit rate experiments in the previous section. As the percentage of wake up cycles increase, the amount of energy lost also increases. A 100 % duty cycle gives the highest energy lost since the CH in this case does not enter the dormant state at all.

These experiments prove that duty cycling according to smartphone density is an effective solution to reduce energy

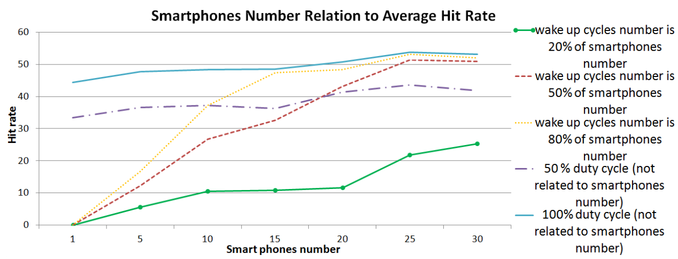

Fig. 12. Wake up cycles' number in relation to hit rate

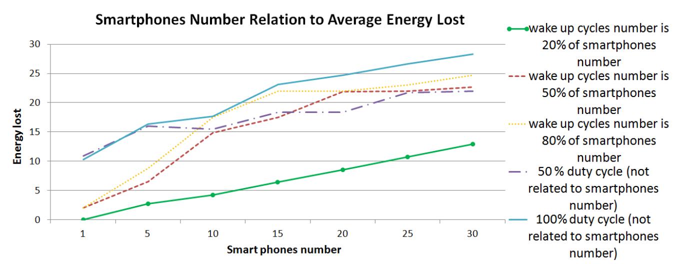

Fig. 13. Wake up cycles' number in relation to energy lost

loss without compromising the hit rate. We can easily conclude that our local SD protocol fulfils its two main objectives which is serving user requests without sensor's battery depletion.

V. Conclusion

SD can be a challenging task in the field of IoT due to the challenges imposed by the dynamic nature of WSNs and limitations of the sensors in such networks, whether in terms of storage capacity or power consumption. Previous research was focusing on solutions using a gateway to publish sensor's data. However, as presented, gateways introduce a new set of challenges in terms of scalability and robustness.

This paper presented a local SD protocol for hierarchical network architecture that is both energy efficient and can provide a high hit rate. The large IoT network was divided into small areas, each consisting of some sensors and their elected CH. The highest level of the heirarchy consisted of area routers that were responsible for multiple areas. In order to achieve the best hit rate, multi-hopping can be enabled either through neighboring CHs or through area routers. Moreover, duty cycles of the network nodes can be adjusted according to the smarphone density in each area. Numerous experiments were conducted to evaluate the efficiency of the protocol proposed and it was actually proven to achieve its goals.

In the future, we are planning to implement a hardware prototype of this protocol in order to evaluate its robustness on a real network. Through this implementation, we are also looking forward to adding more features that can enhance the energy consumption or hit rate of user requests. Furthermore, this protocol can be extended to achieve global SD.

References

[1] J. Jin, J. Gubbi, S. Marusic and M. Palaniswami, *An Information Framework for Creating a Smart City Through Internet of Things*. IEEE Internet of Things Journal, vol. 1, no. 2, pp. 112-121, April 2014.

[2] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. R.Biswas and K. Moessner, *Enabling Smart Cities through a Cognitive Management Framework for the Internet of Things*. IEEE Communications Magazine, vol. 51, no. 6, pp. 102-111, June 2013.

[3] V. Foteinos, D. Kelaidonis, G. Poulios, P. Vlacheas, V. Stavroulaki and P. Demestichas, *Cognitive Management for the Internet of Things, A Framework for Enabling Autonomous Applications*. IEEE Vehicular Technology Magazine, vol. 8, no. 4, pp. 90-99, December 2013.

[4] A. Moraru, C. Fortuna and D. Mladenic, *A System for Publishing Sensor Data on the Semantic Web*. ITI 2011 33rd International Conference on Information Technology Interfaces, pp. 199-204, 27-30 June, 2011.

[5] A. Devaraju, S. Hoh and M. Hartley, *A Context Gathering Framework for Context-Aware Mobile Solutions*. The 4th International Conference on Mobile Technology, Applications, and Systems, pp. 39-46, 10 September 2007.

[6] F. Ramparany, R. Poortinga, M. Stikic, J. Schmalenstroer and T. Prante, *An open context information management infrastructure the IST-amigo project*. IEEE 3rd IET International Conference on Intelligent Environments, pp.398 - 403, 2007.

[7] T. A.Butt, I. Phillips, L. Guan and G. Oikonomou, *Adaptive and Context-aware Service Discovery for the Internet of Things*. Internet of Things, Smart Spaces, and Next Generation Networking, vol. 8121, pp. 36-47, 2013.

[8] S. Cirani, L. Davoli, G. Ferrari, R. Lone, P. Medagliani, M. Picone, and L. Veltri, *A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things*. IEEE, Internet of Things Journal, vol. 1, no. 5, pp. 508-521, October 2014.

[9] H. Luo, L. Liu and Y. Sun, *Semantics-based Service Mining Method in Wireless Sensor Networks*. Seventh International Conference on Mobile Ad-hoc and Sensor Networks, pp. 115-121, 2011.

[10] H. Wirtz and K. Wehrle, *Opening the Loops - Towards Semantic, Information-centric Networking in the Internet of Things*. 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pp. 18-24, 2013.

[11] C. Bormann, A. P.Castellani and Z. Shelby, *CoAP: An Application Protocol for Billions of Tiny Internet Nodes*. IEEE, Internet Computing, vol. 16, no. 2, pp. 62-67, March/April 2012.

[12] S. Li, G. Oikonomou, T. Tryfonas, T. M.Chen, and L. D.Xu, *A Distributed Consensus Algorithm for Decision Making in Service-Oriented Internet of Things*. IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1461-1468, May 2014.

[13] J. Li, H. Wang, S. U.Khan, Q. Li and A. Y.Zomaya, *A Fully Distributed Scheme for Discovery of Semantic Relationships*. IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 457-469, October-December 2013.

[14] Z. Sheng, S. Yang, Y. Yu, A. V.Vasilakos, J. A.Mccann and K. K.Leung, *A Survey on the IETF Protocol Suite for the Internet Of Things: Standards, Challenges, and Opportunities*. IEEE Wireless Communications, vol. 20, no. 6, pp. 91-98, December 2013.

[15] D. Pfisterer, K. Rmer, D. Bimschas, O. Kleine,R. Mietz, C. Truong, H. Hasemann, A. Krller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant and R. Richardson, *SPITFIRE: Toward a Semantic Web of Things*. IEEE Communications Magazine, vol. 49, no. 11, pp. 40-48, November 2011.

[16] INetSimpleBattery class definition in OMNet++, *http://omnetpp.org/doc/inet/api-current/doxy/class_inet_simple_battery.html*

[17] Sensor Datasheet, *https://www.sparkfun.com/datasheets/Sensors/SHT1x_datasheet.pdf*