Neighbor Discovery Algorithms for Friendship Establishment in the Social Internet of Things

Roberto Girau*, Salvatore Martis†, Luigi Atzori‡
Dept. of Electrical and Electronic Engineering
University of Cagliari
Cagliari, Italy
*roberto.girau@diee.unica.it, †salvatore.martis@diee.unica.it, ‡l.atzori@ieee.org

***Abstract*—In the Social Internet of Things (SIoT), objects establish social relationships to create a social network of devices that fosters inter-object communications by improving network navigability and trustworthiness evaluation. An important process of this model is the neighbors discover that should be implemented by each object to create new relationships. Differently from past neighbors discovery works, in the SIoT scenario, the devices need to discover each other when they are assumed to be already able to communicate to the Internet and additionally they do not have to set up direct device-to-device communications after the discovery. These aspects call for the definition of new algorithms, which are the major contribution of this paper. Specifically, three solutions are proposed: the first one relies on the channel scanning; the second one assumes that channel scanning is not possible and makes use of the device localization features; the third one is similar to the second one, but the already existing objects social network is exploited.**

I. Introduction

The IoT (Internet of Things) [1] is an evolution of the Internet that involves a large number of devices that communicate, cooperate and share resources among them to provide integrated and advanced services to the user. The huge number of devices that are expected to be connected in the coming years has posed several challenges that need to be faced by the research community in the near future. This myriad of objects will involve heterogeneous technologies, standards and unbalanced capabilities in terms of processing, communication, and energy availability and will produce an enormous amount of data difficult to be analyzed with several problems of trustworthiness and privacy. To address these issues, the idea of using the technologies and results related to the field of social networking in the IoT to enable objects to autonomously set their own social relations is gaining popularity in recent years [2]. The main reason is that objects augmented with social capabilities can improve search, selection and composition of services and information, provided by the communities of objects taking part to the IoT. According to the specific object socialization model of SIoT (Social Internet of Things) defined in [3], the objects may create different kinds of social relationships. The relation "Parent Object Relationship" (POR) is defined between similar objects, built in the same period by the same manufacturer (the family's role is played by the production batch). In addition, objects can establish a relationship of co-location (C-LOR) or a relationship of co-Work (C-WOR), as humans do when they share personal places (i.e. the house) or work places. Another type of relationship is defined for objects owned by the same user (mobile phones, game consoles, etc.) and called "Ownership Object Relationship" (OOR). The last type of relationship is created when objects come into contact, sporadically or continuously, for reasons purely related to the relations between their owners (for example, devices/sensors belonging to friends); this relation is called "Social Object Relationship" (SOR). To this, any object is equipped with a module that implements the relationship management procedures. These procedures should identify the required events for the establishment of friendship of the types mentioned before. These procedures should be performed partially in the object itself and partially in the cloud where the virtual counterpart of it is running (the Social Virtual Object, SVO).

To implement the depicted objects' socialization algorithm, there is the need for a procedure for the discovery of neighbors, i.e., Neighbor Discovery (ND) algorithm. This should be effective in identifying for how long and how many times a couple of objects have been close to each other. The ND algorithm is the first step for the establishment of friendship and should allow each object to discover all its neighbors and communicate its presence to every neighbor on the network [4]. Differently from past neighbors discovery works, in the SIoT scenario, the devices need to discover each other when they are assumed to be already able to communicate to the Internet and additionally they do not have to set up direct device-to-device communications after the discovery. These aspects call for the definition of new algorithms, which are the major contribution of this paper. Specifically, three solutions are proposed: the first one relies on the channel scanning; the second one assumes that channel scanning is not possible and makes use of the device localization features; the third one is similar to the second one, but the already existing objects social network is exploited. These algorithm are also integrated in the Lysis platform [5], which is a first implementation of SIoT paradigm that exploits virtual entities in cloud as avatars of physical objects.

978-1-5090-4130-5/16/$31.00 ©2016 IEEE

The paper is organized as follows. In Section II we describes the previous works on Neighbor Discovery algorithms and the related guidelines that drove our design of new algorithms with respect to existing solutions, while in Section III we show our solution and in Section IV a performance evaluation of the proposed models. Finally in Section V, we draw final remarks.

II. RELATED WORK

There is significant number of works on ND as part of ad-hoc, opportunistic and vehicular networks. The following subsections, present the algorithms already proposed in the past.

A. Neighbor discovery in IoT

The Neighbor Discovery is traditionally analyzed in wireless sensor networks area (WSN) to solve energy problems [6]. During the deployment phase, it is evident that if the devices were kept always active and listening, it would waste a considerable amount of energy just to accomplish the network topology. So, initially, the research has focused on how to save energy in the process of discovery by finding a compromise with the latency (the time it takes to be aware of the presence of another device). The alternating between sleep and awake states of the radio interfaces has been the focus of several strategies for neighbor discovery.

In the definition of Neighbor Discovery strategies we must address several issues. Firstly, the ability of an algorithm to recognize the presence of other nearby devices. It involves a continuous adapting of the radio resources in order to find the neighbors while staying within a time frame and under contact conditions that vary dynamically. Conversely, figuring out if there are no devices in the vicinity saves energy instead of wasting it in useless discovery operations. Secondly, the use of pattern recognition methods for mobility allows for capturing spatial and temporal characteristics [7][8]. Other models use metrics of popularity of the visited places, social behavior, tagging places such as roads, schools, airports etc. to better identify mobility patterns [9][10]. Finally, the problem of the acquisition of information related to future encounters not only in terms of storage but also in terms of understanding them, requires more efforts in research to allow for a greater degree of predictability of the movements and allow for better management of energy resources. Keeping in mind these issues, the devices can be kept operational for a longer time so as to increase the time of better planned communications D2D .

Generally, in ND protocols, all the time-related parameters (arrival/departure time, contact time, inter-contact time) can be managed either synchronously or asynchronously. In ZebraNet [11] the GPS time is used to allow for synchronization of the nodes. In [12] a random sensing technique based on time synchronized using a Markov chain is proposed to optimize scheduling between active and dormant states. In *Recursive Binary Time Partitioning* (RBTP), the authors claim to reduce the latency of discovery between smartphones using the NTP protocol for synchronization. In *WizSync* [13], IoT devices with ZigBee interface can take advantage of the overlap in the radio bands at 2.4 GHz of unlicensed spectrum to achieve synchronization via Wi-Fi beacons. The asynchronous protocols do not require the presence of an accurate time reference to do the discovery between neighboring devices. Usually, they use indirect requests or exploit the overlapping of the time slots to activate the discovery. In *Sparse Topology Energy Management* [14], the authors propose the use of two radio interfaces operating in parallel and at different frequencies, one for the communication channel and one for the wake-up signals. In [15], it is brought an improvement to the concept of "Wake on Wireless" via a platform that uses Wi-Fi, Bluetooth and ZigBee in various combinations on the assumption that low-power radio systems combined with other high-power ones are able to optimize the discovery and communication. In ZiFi [16] signs of interference are used in ZigBee to deduce the presence of Wi-Fi access point. In *Random Asynchronous Wake-up* (RWA) [17], randomization in dense scenarios allows for the maximization of the probability that the nodes can see each other. Choi et al [18] presented a hierarchical and adaptive approach based on difference sets that allow for a choice among different levels of energy efficiency. Bacht et al [19] with *Searchlight*, has proposed a new protocol which can obtain discovery by delay constraints through deterministic search sequentially the time slots. In [20], assuming that the contact time follows a power law distribution, authors show that the probability of contact failure does not depend on the reducing the duty cycle if $T_{ON} \geq T_{OFF}$ and $\tau \geq 2(T - T_{ON})$, where T, T_{oN} and T_{OFF} are respectively the period, the sleep and awake time, and τ is the minimum duration of contact. Other algorithms instead take into account the knowledge of the movements of nodes to decide the schedule and can be based on arrival times and on the encounter rate between nodes. In [21], it is defined an efficient communication protocol that leverages the knowledge of repetitive and predictable movements such as those of the public buses. The authors show that if the spatial distribution of the nodes has a guaranteed minimum distance of separation, then the probability of meeting failure is zero. In [22] instead, the authors introduces a framework for the energy management based on knowledge of contacts between IoT objects, through an approach on "zero knowledge" which adopts a beacon strategy for synchronous systems. In [23], the *Reinforcement Learning* (RL) is exploited to allow for collecting online data on mobility patterns. *Resource Aware Data Accumulation* (RADA)[24] leverages a framework based on the Q-Learning for efficient discovery in terms of energy with fixed devices capable of learning the presence of mobile devices. *Sensor Node Initiated Probing for Rush Hours* (SNIP-RH) [25] proposes an algorithm that exploits the knowledge about the rush hours of the day to concentrate efforts and energy in the planning of discovery when it is more likely the interaction between objects.

B. Key Aspects of Neighbor Discovery Algorithms

By analyzing the state of the art on ND algorithms, we are able to identify the major features that characterize the background in which the above mentioned algorithms are thought (Fig. 1(a)):

- Radio Channel Scan: all devices are able to (and have to) scan the radio channel for neighbor discovery.
- ND for device communication: the goal of ND algorithms is always a reliable and durable communication between the discovered devices through the radio channel.
- No middleware support: applications at higher level are not allowed for accessing the list of retrieved neighbor devices.
- Low level access to radio interfaces: ND algorithms are thought to work in the radio chipset firmwares, and usually at MAC level.

In SIoT applications, we have to face other issues related to a different scenario (Fig. 1(b)):

- Heterogeneous communication interfaces: the IoT includes a wide variety of devices with different hardware components and interfaces often using different radio channels and protocols.
- D2D communication not required: the only requirement is the communication to the Internet in order to access to the IoT virtual counterparts.
- Middleware support: the list of neighbor devices has to be known through middleware APIs to evaluate the friendship accomplishment.
- Standard Control of radio interface: low level access to the radio interfaces is not required.

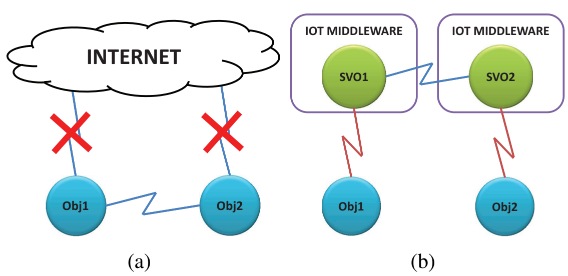

Fig. 1. Different scenarios of applicability of ND algorithms: standard scenario in ad-hoc, opportunistic and vehicular networks (a); scenario related to friendship establishment in SIoT (b)

A consequence of these new requirements, a new approach has to be adopted in order to design new algorithms needed for friendship detection in the SIoT scenario.

III. Neighbor Discovery Algorithms in SIoT

As already mentioned, we consider the Lysis solution for the implementation of the SIoT. Accordingly, every device in Lysis has an autonomous virtual counterpart in the cloud named Social Virtual Object (SVO). When two SVOs have established a relationship, they can easily exchange information and resources. As described in Section I, two relationship types (i.e., SOR and C-WOR) depend on the frequency and duration of meeting events between objects. As a consequence, the ND procedure becomes of primary importance. Unlike in opportunistic networks where the ND is necessary as a first step before they can communicate, the ND in SIoT is aimed at the construction of the social graph when the devices are already capable to communicate to the Internet. Among the devices to be considered, we have to make a distinction depending on the level of access to the hardware granted by the respective firmware. To one category belong the devices that may not have either the permissions to perform a scan of nearby devices (i.e., Apple IOS devices) or to know their own MAC address (i.e., devices with Android OS version 6). This category also includes the devices that do not have wireless interfaces, but can provide information about their location using GPS systems. The other category of devices is that of those that have complete control of the wireless interface. To consider the scenarios with these two kinds devices, we propose three ND algorithms aimed at creating social relationships between objects.

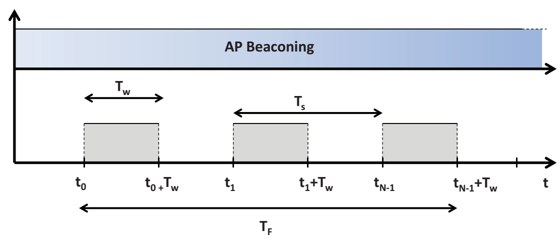

Fig. 2. Time parameters involved in the neighbor discovery process needed for friendship creation

Despite in our scenario the physical devices do not have to scan interfaces to find neighbors, the SVOs make the actual decisions about scan frequency and duration, and implement the rules for the creation of the relationships. The friendship creation algorithm executed by the SVO requires two devices to be in continuous visibility for a minimum time T_f, after which each SVO asks to the other to start a new friendship. To verify this condition, the visibility sampling performed by the radio interfaces must be at the most equal to a value of T_s, which is the inter-scan time. Therefore, it seems clear that we must take into account these constraints to evaluate the duty cycle of the radio interfaces. The inter-scan time T_s, is determined by the friendship rules, and in the presence of nearby devices must be

$$T_s = \frac{T_F}{N} \quad (1)$$

where N is the number of meetings (samples) within the friendship time T_F. For example, if the SOR relationship requires 30 minutes of visibility to be established and six encounters, then it has to be $T_s = 5$ minutes. As shown in Fig 2, the duration of the discovery is T_w, and depends on the radio media (for WiFi it is about 1 second, for Bluetooth it is up to 10 seconds) and usually it is negligible compared to friendship time (30 minutes) but affects the inter-scan time since it has to be $T_s \geq T_w$. In the following three subsections we present the proposed ND algorithms. The local neighbor discovery aims at exploiting the presence of wireless reference points to detect encounters and visibility between objects. The discovery by

PubSub aims at allowing objects to share information about the position through systems of publish/subscribe in order to detect ephemeral events of collocation. The same goal is pursued by the discovery through social alerting, which exploits the social graph to spread information about the objects' positions.

A. Local neighbor discovery

The Local Neighbor Discovery (LND) algorithm can be implemented by devices able to scan a radio channel to look for other devices that transmit beacons, such as: WiFi Access Points, Bluetooth beacons for indoor navigation or for advertisement, mobile devices in hotspot mode. The basic approach behind this algorithm is the use of one of these devices as *reference point* (RP). The objective is then to detect

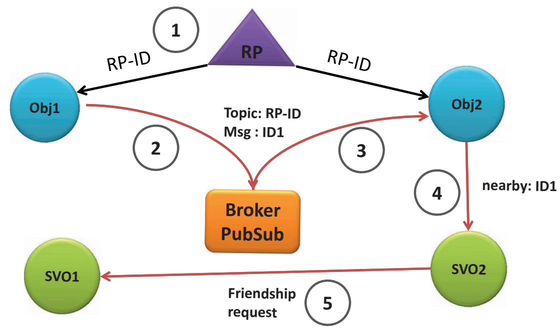

Fig. 3. Local Neighbor Discovery: the encounter detection is implemented on the physical devices

the situations where two devices are under the same radio coverage of the same RP. Such reference point devices do not need to be registered to the SIoT platform, since the RP's ID (e.g., MAC Address and SSID) is achievable in the beacon message and it is not required an object-RP interaction. In such a way, by means of this approach it is possible to exploits the high concentration of RPs especially in the urban environment. To better explain the algorithm we consider the scenario depicted in Fig. 3 and described in the following. Let us consider the case of two devices, Obj1 and Obj2, both registered on the Lysis platform. These devices have unique IDs generated at the time they have been associated with the respective SVOs (say ID1 and ID2). Let us suppose Obj1 enters into the radio range of an reference point RP which is identified by the MAC address (RP-ID) (1) and where Obj2 already is. Obj1 performs a subscribe to topic "RP-ID" and performs a publish in the same topic stating its SVO's ID (ID1) (2). As long as Obj1 is in the RP coverage area, it performs the same publishing operation each T_s decided by its SVO (SVO1) and represents the minimum time between two samples so that these samples represent the same encounter and not two distinct encounters. Every time Obj1 publishes this message, Obj2 who has subscribed the same topic, receives a message with SVO1's ID (3) and sends a message to its SVO (SVO2) stating the identity of the neighbor (ID1) (4). After N meetings SVO2 proceeds to send a friendship request to SVO1 (5). The process is dual, and therefore SVO1 after N meetings with SVO2 sends the friendship request to SVO2. Requests are then accepted and the two SVOs create a relationship.

This process requires a passive participation of the devices, which have only to perform the sensing. However, the chances to perform the discovery can be increased by enabling the beaconing on devices (if they are able to), which will become a reference for the neighboring devices. In this algorithm, the crucial point is the discriminating threshold of the signal strength and the inter-scan time of the interfaces. A high threshold will reduce the number of seen objects, while a low threshold may erroneously lead to the conclusion that two objects are close to each other while being at distance of up to a hundred meters, especially in networks with wide coverage.

Beacon duration and frequency are the key elements only in some situations. Indeed, if some reference points are present, then it is not necessary that the devices start to transmit beacons, given that there is already a radio reference point. The beaconing should be activated in the absence of other reference points. The alternating of beaconing/listening states may use algorithms already present in the literature to decide the duration and frequency of beaconing so as to ensure maximum probability of encounters.

B. Discovery by PubSub Alerting

In this scenario, we consider devices that cannot scan radio interfaces for hardware reasons (for example devices with GSM, UMTS, LTE interfaces) or for software permission reasons (for example devices with Apple's IOS operating systems). In this case, the approximate location via GPS or Wifi has to be known. The PubSub-Alerting Discovery (PSAD) algorithm uses an alert system that relies on the Publish/Subscribe protocols (e.g., MQTT) when registering in a place. For this reason, it is needed a common repository of

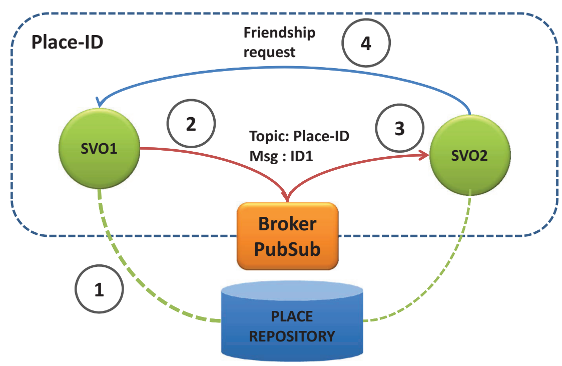

Fig. 4. Discovery by PubSub Alerting

places where SVOs can find the right correspondence between GPS coordinates and the places they have to register in. Every time the physical device detects its position, it sends it to its virtual counterpart. The sampling frequency and the sending of the position are adaptive depending on the speed, in such a way that the positions are evaluated only when the object is stationary or almost-stationary. The discovery algorithm is performed entirely in the cloud virtual counterparts.

In Fig. 4, the SVO1 (virtualization of the object Obj1) receives the position from its physical device and retrieves the related Place-id from the place repository (1). Obj2 is already in the same place and the related SVO (SVO2) has already subscribed the topic that refers to this place. At this point, SVO1 performs a subscribe to the topic "Place-ID" and a publish in the same topic stating its own ID (ID1) (2). As long as Obj1 is in this place, SVO1 sends a publish each T_s interval, whose publish messages are received by SVO2 (3). After N received messages with ID1 as ID, SVO2 sends a friendship request to SVO1 (4). The process is dual, and therefore, after N messages with ID2, SVO1 sends the friendship request to SVO2. Requests are accepted and SVO1 and SVO2 create a relationship.

C. Discovery by Social Alerting

Also in this scenario, the devices are not required to scan the radio channels but need to be able to determine their positions. Unlike the previous case that uses a central element (the PubSub broker), the Social-Alerting Discovery (SAD) algorithm exploits the social graph to spread the alerts about registrations in a place in a gossip-like way. By word of mouth, through friends and friends of friends, two nodes that do not know each others can communicate. In Fig. 5, SVO1 and SVO2 are virtual objects of devices in the same place, which has ID retrievable from place repository (1). The place where the objects are in, is added in the profile of each SVO of both fixed or mobile devices. In the mobile device case, the descriptor is added or updated only if the object is not moving. SVO1 sends an alert message saying that it is in the place with identifier "Place-ID" to its friends (2) and forwarded by SVO3 (3) and SVO4 (4) to SVO2. In fact, SVO3 is friend with SVO1 and SVO4, whereas SVO2 is friend with SVO4. After N alerts, SVO2 sends a friendship request to SVO1 (5). Also in this algorithm, when the requests are sent through both directions the friendship relationship is created.

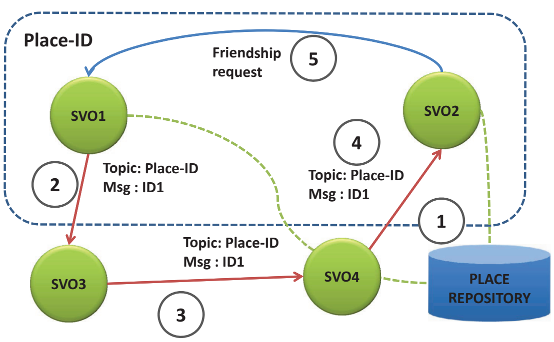

Fig. 5. Discovery by Social Alerting

IV. Experimental Evaluation

A. Simulation Setup

To perform our experimental analysis, we would need mobility traces of a large number of objects. Since real data of this type are not available at the present, we resorted on a mobility model called Small World In Motion (SWIM) [26] to generate synthetic traces. The outputs of the SWIM model are traces of the position and encounter events of humans. In this paper, instead, we are interested in the mobility of things and consequently, we assume that each user brings a device with WiFI interface for testing the first algorithm, or a GPS system for the other two algorithms. To evaluate the performance of the first algorithm, we considered an area of 17000 sq.m. with different numbers of reference points (WiFi access point in the simulation) from 10 to 30 randomly scattered. We assumed a coverage radius of 30 meters for each reference point. For the other two algorithms, we consider the same area without reference points.

B. Simulation Results

To evaluate the LND algorithm, we want to get the number of encounters that the algorithm has been able to detect compared to the total number of real encounters, whose ratio is named Hit Ratio (HR). In each simulation, we varied the scan duration T_w , the total number of access points in the area and we set two different values for the inter-scan time T_s of 60 and 300 seconds resulting into two groups of curves. In Fig. 6

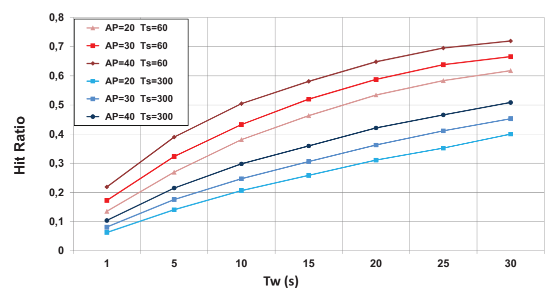

Fig. 6. Hit Ratio when using LND algorithm

the results of the simulations are shown. As expected the HR grows with the scan duration T_w and the total number of APs. Despite it is true that there are fewer detected meetings with a longer inter-scan time, in this case, devices have to be closer for longer unveiling a higher probability of common interests. Since the APs' density cannot be set arbitrarily in the real world, the only variable parameter which affects the number of discovered neighbors is the inter-scan time and, with less influence, the scan duration.

The other two algorithms are evaluated by using SWIM traces and by simulating GPS coordinates acquisition. From devices' coordinates we inferred the places which the devices visited. In the simulation we considered the same 17000 sq.m area of the LND algorithm positioned in the campus of the Faculty of Engineering in Cagliari, we used Google places service as place repository, and accordingly we got six places in the considered area. In the simulation we varied the waiting time T_{wait} needed by the SVO to understand if the related device is moving or steady and for the SAD we considered the cases of 2 and 3 hops as time-to-live of the alert messages (involving friends of friends in the 2 hop case or more in the 3 hop one) over a social graph of 100 nodes and average node

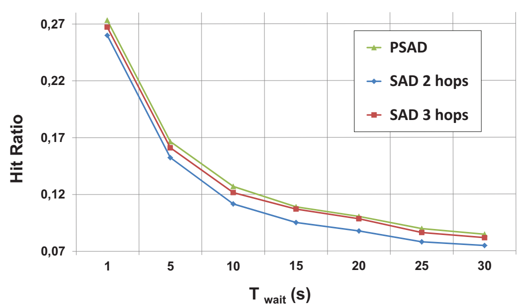

Fig. 7. Hit Ratio exploiting the devices' positions

distance equal to 4. In Fig. 7 we can see that at higher T_{wait} corresponds poorer performance in detecting neighbor devices. The two algorithms perform quite the same so as we can infer that the best choice in this scenario is given by SAD with 2 hops since there isn't a central node like the PubSub broker and the messages' overhead is lower. As general observation, the SAD algorithm is not to be thought as alternative for the LND algorithm because they refer to different scenarios. On the contrary, they should be considered as complementary approaches in order to detect as many devices as possible.

V. Conclusion

In this paper we have presented three algorithms for neighbor discovery needed for detecting new relationships between objects in the Social Internet of Things. This new paradigm, with the intensive use of virtual counterparts in the cloud, demands for new solutions in order to detect as many neighbors as possible without the constrain of direct device-to-device communications. One of the proposed algorithms relies on the scan of the radio channel, whereas the other two assume that this is not possible and make use of the device localization algorithm.

As future work, we are planning of integrate these algorithms in Lysis platform in order to evaluate them in real scenarios.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[2] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between internet of things and social networks: Review and research challenges," *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 206–215, 2014.

[3] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[4] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, and A. Gluhak, "Neighbor discovery for opportunistic networking in internet of things scenarios: A survey," *Access, IEEE*, vol. 3, pp. 1101–1131, 2015.

[5]

[6] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2001, pp. 137–145.

[7] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of intercontact times between mobile devices," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 10, pp. 1377–1390, 2010.

[8] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM transactions on networking (TON)*, vol. 19, no. 3, pp. 630–643, 2011.

[9] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: a location predictor on trajectory pattern mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 637–646.

[10] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "Nextplace: a spatio-temporal prediction framework for pervasive systems," in *Pervasive computing*. Springer, 2011, pp. 152–169.

[11] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in zebranet," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 227–238.

[12] G. Ghidini and S. K. Das, "An energy-efficient markov chain-based randomized duty cycling scheme for wireless sensor networks," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011, pp. 67–76.

[13] T. Hao, R. Zhou, G. Xing, M. W. Mutka, and J. Chen, "Wizsync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 6, pp. 1379–1392, 2014.

[14] C. Schurgers, V. Tsiatsis, and M. B. Srivastava, "Stem: Topology management for energy efficient sensor networks," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3. IEEE, 2002, pp. 3–1099.

[15] T. Pering, V. Raghunathan, and R. Want, "Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup," in *VLSI Design, 2005. 18th International Conference on*. IEEE, 2005, pp. 774–779.

[16] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "Zifi: wireless lan discovery via zigbee interference signatures," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 49–60.

[17] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar, "Random asynchronous wakeup protocol for sensor networks," in *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*. IEEE, 2004, pp. 710–717.

[18] B. J. Choi and X. S. Shen, "Adaptive asynchronous sleep scheduling protocols for delay tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 9, pp. 1283–1296, 2011.

[19] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 185–196.

[20] H. Zhou, H. Zhao, and J. Chen, "Energy saving and network connectivity tradeoff in opportunistic mobile networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 524–529.

[21] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 129–145.

[22] H. Jun, M. H. Ammar, and E. W. Zegura, "Power management in delay tolerant networks: a framework and knowledge-based mechanisms." in *SECON*, vol. 5. Citeseer, 2005, pp. 418–429.

[23] V. Dyo and C. Mascolo, "Efficient node discovery in mobile wireless sensor networks," in *Distributed Computing in Sensor Systems*. Springer, 2008, pp. 478–485.

[24] K. Shah, M. Di Francesco, G. Anastasi, and M. Kumar, "A framework for resource-aware data accumulation in sparse wireless sensor networks," *Computer Communications*, vol. 34, no. 17, pp. 2094–2103, 2011.

[25] X. Wu, K. N. Brown, and C. J. Sreenan, "Exploiting rush hours for energy-efficient contact probing in opportunistic data collection," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*. IEEE, 2011, pp. 240–247.

[26] A. Mei and J. Stefa, "Swim: A simple model to generate small mobile worlds," in *INFOCOM 2009, IEEE*, 2009, pp. 2106 –2113.