# OppNet: Enabling Citizen-Centric Urban IoT Data Collection Through Opportunistic Connectivity Service

Fengrui Shi[1], Usman Adeel[2], Evangelos Theodoridis[2], Mo Haghighi[2], and Julie McCann[1]

[1]fengrui.shi14, j.mccann@imperial.ac.uk, Department of Computing, Imperial College London
[2]firstname.lastname@intel.com, Intel Labs Europe

*Abstract*—Urban IoT data collection is challenging due to the limitations of the fixed sensing infrastructures. Instead of transmitting data directly through expensive cellular networks, citizen-centric data collection scheme through opportunistic network takes advantage of human mobility as well as cheap WiFi and D2D communication. In this paper, we present OppNet, which implements a context aware data forwarding algorithm and fills the gap between theoretical modelling of opportunistic networking and real deployment of citizen-centric data collection system. According to the results from a 3-day real-life experiment, OppNet shows consistent performance in terms of number of hops and time delay. Moreover, the underlying social structure can be clearly identified by analysing social contact data collected through OppNet.

*Keywords*-urban sensing; smart city; IoT platform; citizen-centric service; opportunistic network

## I. INTRODUCTION

The evolution of cities, especially regarding urban sensing systems [2] [4], has enabled smart city functions such as urban infrastructure surveillance and control of the physical world. Sensing infrastructures are being deployed to perform tasks such as water leak detection and air quality monitoring at city scale. For these approaches to be effective, smart city [5] platforms have to be deployed to interact with sensing infrastructures and transmission of potentially huge amount of data to the city platform is one of the key challenges. Although some sensing devices are able to connect to the Internet directly, this may not be feasible due to high costs or low communication coverage. For instance, cellular data communication is expensive for large volume of sensor data and WiFi connections are not always available.

Opportunistic communication paradigm [3], which employs the ubiquity of mobile phones to collect and relay sensor data through Device-to-Device(D2D) communication channels is one approach to overcome some of the aforementioned problems. In urban sensing scenarios, such citizen-centric services enable mobile phones to relay data and thus bridge the connectivity gap between the sensors and the platform.

Recently, many solutions have been proposed for mobile sensing and data collection with mobile phones. Some data relaying schemes are aimed to fill the connectivity gap for sensor infrastructures [6] [7]. In addition, in order to fully leverage the communication opportunities of mobile devices, one has to optimize the network underlying the system. [8] utilizes stochastic optimization to make routing decisions in the network. While other work [9] considers predicting contact opportunities using statistical inference models. [10] also combines economic model when designing the routing algorithms. Game theory and auction-based approaches are discussed in [11]. To evaluate these approaches, there is a need of a city wide testbed for performing extensive experimentation. However, none of the work considers deploying a working testbed integrated with an urban IoT platform.

In this paper, we implement the first ever opportunistic network testbed for a large scale smart city platform OrganiCity [1]. The system utilizes the mobility and computing power of mobile phones to collect data from urban sensing infrastructures and relay data to the smart city platform according to a context aware routing algorithm. For brevity we call our system OppNet (**Opp**ortunistic **Net**working connectivity service). To our knowledge, it is the first time that an opportunistic mobile sensing and data collection system has been developed and integrated into a large-scale urban IoT platform. The contributions for this paper include:

- A testbed OppNet, which is an opportunistic mobile sensing and data collection system integrated with an urban IoT and sensing platform - OrganiCity, has been deployed.
- A context awareness opportunistic data forwarding scheme has been proposed. This algorithm makes decisions by combing Backpressure routing and the availability of resources in the mobile devices.
- A real-life experiment has been performed to evaluate the performance of OppNet.

### A. Paper Organization

The next section presents an overview of the OppNet system, including its integration with the urban sensing platform-OrganiCity. Section III describes the context aware routing algorithm and the implementation of the OppNet testbed. Section IV presents both the real-time evaluation and the results. Finally, we conclude the paper in Section V.

## II. An Overview of the OppNet System

The OppNet system consists of a middleware currently implemented for Android to interface the urban IoT sensing infrastructure with the smart city platform.

### A. System Architecture

The OppNet collects data either by itself, if it runs a crowdsensing [12] scheme, or from fixed sensing infrastructures by relaying data through cheap and fast D2D connections. Three major components - sensing infrastructure, smart city platform and the OppNet middleware - comprise the whole system for sensing and data collection.

*Sensing infrastructure:* Sensing infrastructure consists of urban IoT devices and mobile sensing devices such as sensors and smartphones. The sensors constantly monitor and record various physical phenomena around them. These devices, generally equipped with short range communication radios, are deployed across the city. Usually such infrastructure is widely deployed across the city in a highly distributed way.

*Smart city platform:* A huge amount of data generated by sensing infrastructures are stored on smart city platforms. The platforms are used to analyse the data and extract valuable information for the management and control of the city. OppNet uses OrganiCity, which is a facility for experimenting smart city services and applications. Fig. 1 provides an overview of Organicity architecture. The facility is exposing an Experimentation as a Service (EaaS) API where experimenters (citizens, researchers, SMEs etc.) can experiment urban services or applications using the resources provided through OrganiCity platform, including a set of tools, APIs, SDKs and libraries. Three cities, Aarhus (Denmark), London (UK) and Santander (Spain), are federated into this platform. As a component belong to the provided tools on OrganiCity, OppNet enables co-creating novel networking infrastructures through citizen-centric urban data collection.
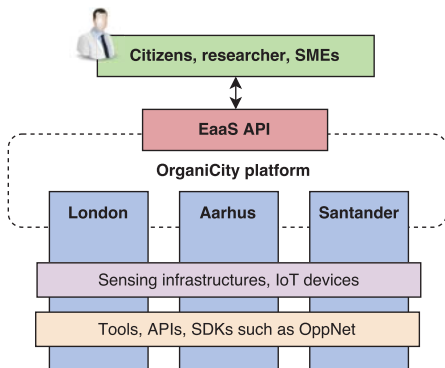


Figure 1.   OrganiCity architecture

*OppNet:* OppNet middleware collects data from sensing infrastructure and send the data to the smart city platform through multi-hop opportunistic routing. Due to the high cost of cellular network, OppNet only uses low cost short range communication such as WiFi and Bluetooth for collection of data and communicating it to the smart city platform. The workflow of OppNet middleware is illustrated in Fig.2.
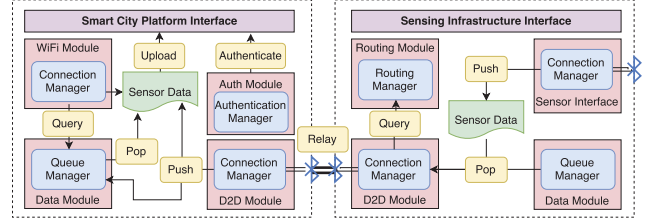


Figure 2.   Workflow of OppNet system

There are three phases in the workflow of OppNet:

- *Data collection*: Fig.2 shows the workflow of a smartphone collecting data from sensing infrastructures and relaying data to another smartphone when they are in contact. To collect data from the sensors, a sensing infrastructure interface is used to identify sensors which have access to the OppNet. Sensor data is transferred from sensors to OppNet middleware through Bluetooth. Then, the collected sensor data are pushed to the phone storage, which is controlled by the queue manager in the OppNet system.

- *Data relay*: When a phone user moves in the communication range of other phones, the D2D module detects such contact opportunities and asks the routing manager for a decision regarding which and how many data to send. The routing algorithm makes decisions under the context of both devices by considering the current size of the queue in the storage, phone battery level as well as other scenario specific factors. The details of the routing algorithm will be introduced in the next section. The receiving device then pushes the received data to its local queue manager.

- *Data uploading*: After receiving data from another phone, the data is either relayed to another device following the same aforementioned procedure, or uploaded to the smart city platform using low cost communication channels such as WiFi. The WiFi module in OppNet middleware is responsible for discovering such opportunities and makes query to the queue manager to retrieve relevant data for uploading. Moreover, in order to control the access to the smart city platform, authentication must be used to identify if certain users can be authorised to upload data to the platform, which is handled by the Authentication module.

## III. OppNet: A Context-Aware Data Collection and Relaying System

### A. Distributed Context-Aware Routing Algorithm

The OppNet system operates a context-aware routing algorithm, which is based on Backpressure [13] and considers the battery level in the algorithm as well.

*Queueing Dynamics:* Each device $x$ maintains a queue $Q_x$ for the sensor data, let $Q_x(t) \geq 0$ be the queue backlog of device $x$ at time $t$. When device $x$ contacts with device $y$, it updates its queue backlog from time $t$ to $t+1$ as follow:

$$Q_x(t+1) = max\{Q_x(t) + (f_{y,x}(t) - f_{x,y}(t)), 0\} \quad (1)$$

where $0 \leq f_{x,y}(t) \leq c_{x,y}(t)$ represents the actual amount of data transmitted from device $x$ to device $y$ at time $t$. This algorithm is derived from basic Backpressure routing algorithm [13]. Such algorithm enables a distributed routing decision making process and thus is suitable for the scenarios of the OppNet system.

*Context Aware Routing:* Since there are some battery hungry operations in the OppNet system such as Bluetooth scanning and wireless communication, the battery level plays an important role in the performance of the system. Therefore, we introduce another component which impacts the routing decisions based on the difference in phone battery levels between two contacting devices to prioritize the phones with higher battery levels as next hop. At time $t$ when device $x$ and $y$ are in contact, device $x$ will make routing decisions based on the following functions:

$$Q_x(t) - Q_y(t) + \alpha(B_y(t) - B_x(t)) \quad (2)$$

where $B_x$ and $B_y$ are the current battery level of both devices and $\alpha$ is the weighting parameter for battery level difference. If the value of Eq.2 is greater than 0, device $x$ will send data to $y$. When $\alpha = 0$, the routing function is the pure backpressure algorithm, while when $\alpha \to \infty$, the algorithm tends to be based on battery level only.

*Incentivisation Scheme:* Based on the routing algorithm in Eq.2, the incentivisation scheme can be derived. The selling price when $x$ sends data to $y$ can be calculated as:

$$Q_x(t) + \alpha(B_{max}(t) - B_x(t)) \quad (3)$$

where $B_max$ is the maximum battery level one device can have, which is 100% here. When $x$ sends data to $y$, $y$ will pay the price as shown in Eq.3. Although $y$ pays for the data this time, it will earn more when it connects to another device since it will have a large queue backlog.

### B. OppNet Testbed Implementation

The OppNet testbed is currently implemented on an Android platform. It employs the Android unsecured rf-comm for Bluetooth, which makes the D2D communication vulnerable to certain attacks, but this design decision enables D2D communication without user interaction, which is an indispensable feature for opportunistic network scenarios.

The Bluetooth Low Energy (BLE) stack with advertising mode enabled is a good alternative, however, there is still a large number of devices not supporting BLE advertising mode. The screenshots of the application are shown in Fig.3.

Fig.3(a) shows the experiment UI, which lists all available experiments in the area where the device resides. The participants are able to choose which experiment to join. In this paper, we implicitly restrict the participants to join the demo experiment only.

Fig.3(b) illustrates the function for authorising the use of this application for OrganiCity platform. It uses OAuth to share the same credentials with OrganiCity platform and thus gives the device permissions to interact with the OrganiCity platform, such as retrieving available experiments, getting user profiles and uploading data to the platform.

Fig.3(c) shows current network status such as Bluetooth communication statistics, WiFi connection status and the rewards the participants have earned.

Fig.3(d) shows the settings page, which not only allows users to turn on/off Bluetooth and WiFi, but also provides two other functions: battery saving and energy awareness. Battery saving provides a low battery operation mode for the application in order to not drain the battery too fast due to some battery hungry operations such as Bluetooth scanning. When the battery saving option is enabled, the Bluetooth scanning interval will be increased and the scanning duration will be decreased, thus consuming less power. The energy awareness option is used to enable the context-aware Backpressure routing protocol introduced before, by considering the current battery level when making routing decisions. It makes the data more likely to flow from devices with low battery levels to high battery levels.

## IV. Evaluating OppNet through Real-life Experiments

We evaluate the OppNet system using real-life experiments. 15 participants have been recruited to perform the 3-day experiment in London. We collect over 9000 data entries and record over 10000 contacts. The evaluation is used to show the performance of the system in real-life scenarios. We are focused on OppNet specific performance metrics such as number of hops and delay. In addition, we also analyse how the data collected from OppNet can help deduce the social structure behind the communication network.

### A. Experiment Settings

The participants are recruited within an area of $1 \ km^2$ and each participant has a *home location*, where they carry out the majority of the activities. If a group of participants have close home locations, they are said to belong to a community. The geographical distribution of the participants are shown in Fig.4. The red dots indicate three locations where the participants are densely located. These locations correspond to the concept of communities and the two red

(a) Experiment      (b) Authorisation
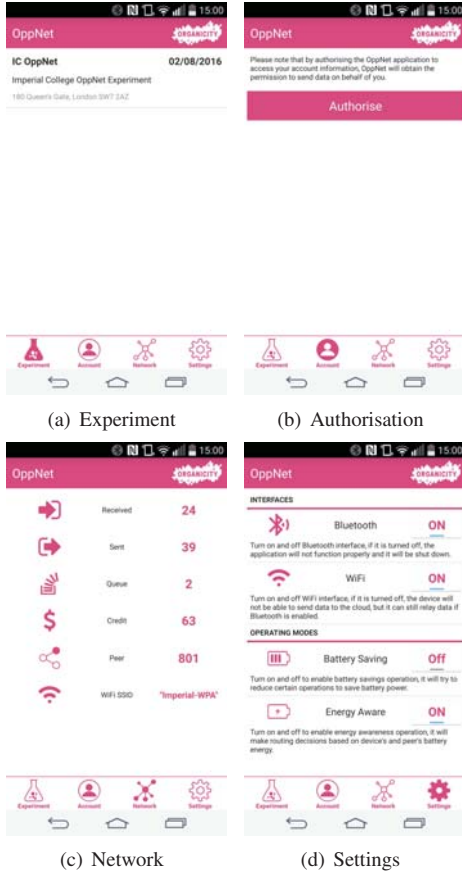
(c) Network      (d) Settings

Figure 3. Screenshots of OppNet smartphone application

dots on the left form one large community. However, we can also show that such social structure can be deduced by simply analysing the data collected from OppNet experiment without knowing the exact home locations of the participants.



Figure 4. Geographical distribution of the participants

During the experiment, all participants carry Android devices with the OppNet application installed. OppNet probes the environment and tries to find nearby devices to connect and relay messages. When the devices find a WiFi connection, OppNet will upload all its current message to the server, which runs an Orion context broker to store the uploaded data. However, since there is good WiFi coverage in the area, in order to simulate the scenario with intermittent WiFi connections and make OppNet relay data, WiFi is only enabled on 8 of the devices. The sensor data is generated locally on mobile phones and the luminous emittance (LUX) data are measured and transmitted.

### B. Datasets

Besides sensor data, the datasets collected by the real-life experiment contain meta data such as delays, paths, as well as contact information. Details of the datasets are shown in Table.I.

Table I
OPPNET DATASETS

| # Data Entries | # Participants | # Contacts | Experiment Duration |
|---|---|---|---|
| 9016 | 15 | 12004 | 50 hours |

### C. Performance metrics

Three performance metrics are selected to evaluate the performance of the OppNet system:

*Number of Hops:* One of the most important metric to evaluate the performance of OppNet is the number of hops the messages take before being received by the server. It reflects not only the delay of the network indirectly, but also the dynamics of the network such as WiFi availability. It is measured by appending hop information in the message being relayed.

*Delay:* There are two types of time delay in OppNet: 1. delay for individual device to relay messages and 2. end-to-end delay of the messages to reach the server. The higher the delay, the less timely the information of the sensor data will be, which makes it a very important metric.

*Contact:* Contact information refers to the interaction among all participating devices. By collecting and processing these data, it offers the insight of certain features of the participants of the experiment such as mobility patterns and social structures.

### D. Results

To better illustrate the results, we present the number of hops in the form of a cumulative distribution function (CDF). The results shown in Fig.5 indicate that 70% of the data are received directly by the server without being relayed through the opportunistic network. Another more than 25% of data are relayed with 2 hops. Less than 5% of all data are relayed by three devices or more. Among them, 2% of the data have 3 hops, 1% have 4 hops and the rest have 5 or 6 hops. The average number of hops for the data to reach the server is

1.33. This value is relatively low compared to large scale urban scenarios due to the fact that good WiFi coverage in the experiment area offers many direct Internet connections.
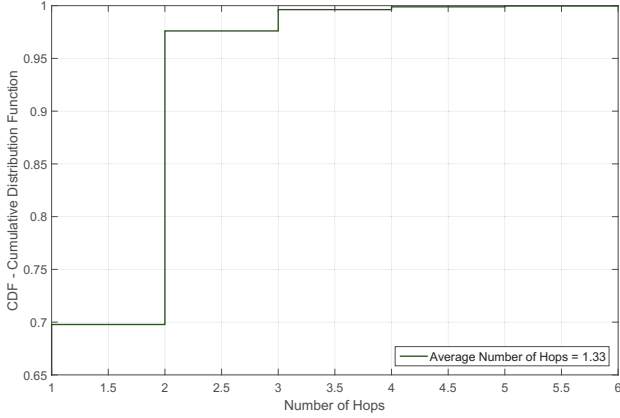


Figure 5. Number of hops it takes before data are received by the server

As for delay, we have two metrics: delay for an individual device to relay data (single hop relay) and end-to-end delay for server to receive data. Single hop relay is the time being taken after each device receives data and before the data can be either relayed to another device or uploaded to the server. Such information can help us understand how long each device holds the data. If the data are held for too long in some devices, it will cause problems such as storage overflow and high end-to-end delay for the data. The end-to-end delay for the server to receive data is thus aggregating all these individual delays by each message, which represents the total time difference when the data are generated and received by the server.

As seen from Fig.6, the CDF plot for end-to-end delay is placed below the plot for individual device delay, which corresponds to the fact that the aggregated delay is greater than individual delay. Fig.6 shows that 70% of all data can be received in short time (<10 minutes), while 10% of all data could be received in more than 10 hours' time, which is a significant delay. This large delay is due to two reasons: 1. some participants took the device with them when they went home, which made the device impossible to communicate with others, and thus accumulated large queue in the phone storage; 2. some device with Android 6 installed has certain requirement that it can only be discovered when the Bluetooth settings page is on. However, some participants unknowingly closed the page and thus made the devices undiscoverable and disconnected from the rest of the participating devices.

As discussed before, the participants can form different communities based on their locations. To show how the data collected from OppNet can reflect such social structure, we draw the social contact graph in Fig.7(a). Each participating device is shown as a bubble. The colour of the bubble
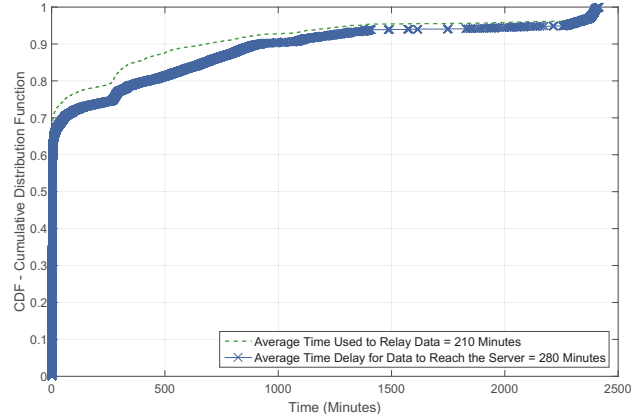


Figure 6. Time delay: *1. single hop relay and 2. end-to-end*

indicates which communities this device belongs to. The size of the bubble shows how many times this device directly contacts the server, which in fact reflects the WiFi availability of each device. The digits inside the bubble shows how many messages have been relayed by each device. The lines connecting the bubbles show the contacts among them with the line width indicating the contact times. The thicker the lines, the more contacts have been made between two devices.

As shown in Fig.7(a), three communities can be deduced from the contact data. The way to identify the communities is to consider contact times. If two participants can be connected with only the thick lines, they belong to the same community. The social structure shown in Fig.7(a) also corresponds to the locations distribution of the participants. The community in green is located far from the other two and thus has no chance talking to the others, which represents the isolated red dot in Fig.4. However, although not closely connected, the other two communities still have some chance to communicate with each other. This shows the situation in Fig.4 that two small communities can form a larger one.

As for WiFi availability, if a device has active WiFi connection, it can upload data to the server and thus has a bigger bubble. Since we only turned on WiFi in 8 of all devices, there are also 8 devices with relatively larger bubbles (>200). Another important observation is that the number of messages relayed by a certain device also correlates with the WiFi availability, with better availability (larger bubble), more messages are also relayed (larger digit). This corresponds to the Backpressure algorithm which routes the messages towards the sinks, which are the devices with active WiFi connections in this case.

## V. CONCLUSION

In this paper, we developed the first OppNet testbed on Android platform. The testbed is integrated with the
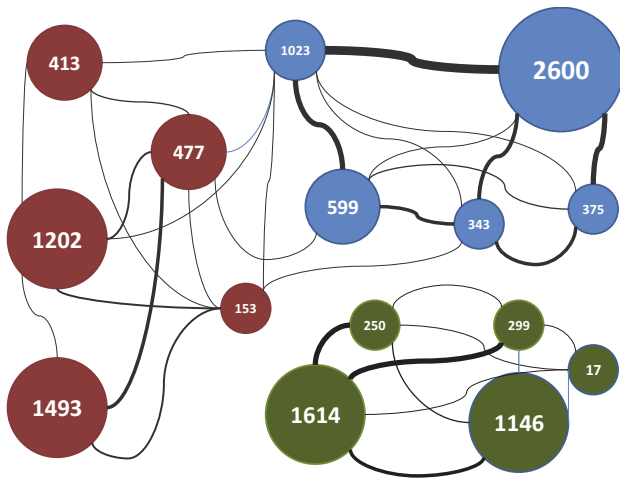
(a) Contact graph of participating devices: *Each bubble represents one participating device, colour indicates community structure, bubble size for WiFi availability, the digits in the bubbles for number of relayed messages and width of lines connecting bubbles for contact times*
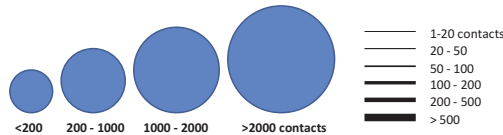


(b) Legend of contact graph

Figure 7.    Contact graph

urban IoT/sensing platform OrganiCity and can be used to perform mobile sensing and data collection tasks for the platform. We proposed a context awareness routing algorithm based on Backpressure and battery level to assist the data forwarding decision making process. A 3-day real-life experiment proves consistent performance of the OppNet system. From the results, it shows that with certain level of WiFi availability, the system can perform the sensing and data collection tasks by incurring only small number of hops (1.33 on average) and 70% of traffic with near real-time end-to-end delay. In the future, the OppNet system will be improved with full integration with OrganiCity. Moreover, with numerous Open Calls beginning for OrganiCity, experiments with unique context will be implemented on OppNet, which would test the performance under various application specific circumstances with different scales.

## REFERENCES

[1] OrganiCity: http://organicity.eu/

[2] Park U and Heidemann J, *Data muling with mobile phones for sensornets*, in Porc. ACM Sensys, 2011, pp. 162175.

[3] Conti M, Giordano S, May M, and Passarella A, *From opportunistic networks to opportunistic computing*, IEEE Commun. Mag., vol. 48, no. 9, pp. 126139, 2010.

[4] Conti M and Kumar M, *Opportunities in opportunistic computing*, Computer, vol. 43, no. 1, pp. 4250, 2010.

[5] OGrady M and OHare G, *How smart is your city?*, Science, vol. 335, no. 6076, pp. 15811582, 2012.

[6] Dunbabin M, Corke P, Vasilescu I, and Rus D, *Data muling over underwater wireless sensor networks using an autonomous underwater vehicle*, in Proc. IEEE ICRA, 2006, pp. 20912098.

[7] Tekdas O, Isler V, Lim J, and Terzis A, *Using mobile robots to harvest data from sensor fields*, IEEE Wireless Commun., vol. 16, no. 1, pp. 2228, 2009.

[8] Yang S, Adeel U, McCann J A. *Selfish mules: Social profit maximization in sparse sensornets using rationally-selfish human relays*. IEEE Journal on Selected Areas in Communications, 2013, 31(6): 1124-1134.

[9] Gao W, Li Q. *Wakeup Scheduling for Energy-Efficient Communication in Opportunistic Mobile Networks*, in IEEE INFOCOM, 2013.

[10] Adeel U, Yang S, McCann J A. *Self-optimizing citizen-centric mobile urban sensing systems*, 11th International Conference on Autonomic Computing (ICAC 14). 2014: 161-167.

[11] Haghighi M, Maraslis K, Tryfonas T, Oikonomou G., Burrows A, Woznowski P, and Piechocki R. *Game Theoretic approach towards Optimal Multi-tasking and Data-distribution in IoT*. IEEE World Forum on Internet of Things (WF-IoT), 2015: 406-411.

[12] Ganti R K, Ye F, Lei H. *Mobile crowdsensing: current state and future challenges*. IEEE Communications Magazine, 2011, 49(11): 32-39.

[13] Ryu J, Ying L, and Shakkottai S. *Back-pressure routing for intermittently connected networks*, in Proc. IEEE Infocom, 2010, pp. 15.