The Computer Journal Advance Access published February 16, 2015

doi:10.1093/comjnl/bxv005

Contact Probing Mechanisms for Opportunistic Sensor Data Collection

Xiuchao Wu, Kenneth N. Brown and Cormac J. Sreenan*

CTVR, Department of Computer Science, University College Cork, Cork City, Ireland
**Corresponding author: cjs@cs.ucc.ie*

In many emerging wireless sensor network scenarios, the use of a fixed infrastructure of base stations for data collection is either infeasible, or prohibitive in terms of deployment and maintenance costs. Instead, we consider the use of mobile devices (i.e. smartphones) carried by people in their daily life to collect data from sensor nodes opportunistically. As the movement of these mobile nodes is, by definition, not controlled for the purpose of data collection, synchronization through contact probing becomes a challenging task, particularly for sensor nodes, which need to be aggressively duty-cycled to conserve energy and achieve long lifetimes. This paper formulates this important problem, providing an analytical solution framework and systematically investigating the effective use of contact probing for opportunistic data collection. We present two new solutions, Sensor Node-Initiated Probing (SNIP) and SNIP-Rush Hours, the latter taking advantage of the temporal locality of human mobility. These schemes are evaluated using numerical analysis and COOJA network simulations, and the results are validated on a small sensor testbed and with the real-world human mobility traces from Nokia MDC Dataset. Our experimental results quantify the relative performance of alternative solutions on sensor node energy consumption and the efficacy of contact probing for data collection, allowing us to offer insights on this important emerging problem.

Keywords: wireless sensor networks; opportunistic data collection; contact probing; human mobility; smartphone

Received 26 September 2014; revised 13 December 2014
Handling editor: Alan Marshall

1. INTRODUCTION

As wireless sensor networks mature, we expect to see long-term deployments for applications such as environmental monitoring, domestic utility meter reading and structural health monitoring. These applications typically involve large numbers of sparsely deployed (static) sensor nodes that report data that are inherently delay tolerant, since the response (if any) requires human intervention over long time scales. For example, analysis of environmental monitoring data is not always urgent, and meter readings for billing purposes can be delayed by weeks. Neighboring nodes in these sparse wireless sensor networks are far away from each other, and typically cannot communicate directly or even indirectly through multi-hop paths. On the other hand, deploying large numbers of fixed base stations (also called sinks) would incur prohibitive costs in terms of deployment, maintenance and data back-haul.

Resource-rich mobile nodes (mobile base stations [1–6] or mobile relays [7, 8]) had been proposed to move around in the deployed area and collect data from sensor nodes. Depending on the application, the mobile nodes can be either part of the external environment [2, 6] or part of the network [4, 5], and their mobility can be either controllable [1, 3] or not [2, 7]. In this paper, we assume that their mobility is not controlled for the purpose of data collection and thus the sensor data are collected opportunistically. Mobile nodes could be specific devices carried by objects (animals, employees, etc.) who move around the deployed area for purposes other than data collection. More interestingly, as illustrated in Fig. 1, they could also be the increasingly ubiquitous smartphones carried by unrelated people who pass through the deployed area in their daily life. In this paper, a smartphone and its user will be referred to as a mobile node.

Under this scenario, smartphones will gather data from sensor nodes automatically and incidentally (without any user intervention and route change). To participate, a smartphone user just needs to run a background application on the

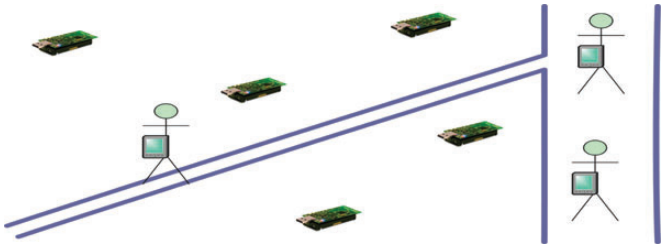

FIGURE 1. Opportunistic data collection.

phone. Here, we assume that smartphone and sensor node can communicate through some low-power radios (IEEE 802.15.4,[1] etc.). Consequently, many smartphone users could be motivated with a very low reward and the total cost of data collection can be reduced significantly through exploiting their uncontrolled mobility. The feasibility, incentive, security and privacy issues that arise in opportunistic data collection with smartphones have been discussed in [9, 10] and these topics are beyond the scope of this paper. Furthermore, the use of smartphones in opportunistic data collection has some of the benefits of adopting mobile base stations discussed in [3, 7], such as the increased network reliability through removing the dependency on base stations, the energy-efficient data collection through reducing the distance from data source to base station and the extended network lifetime through removing hot-spots near the fixed base station. Although opportunistic data collection may increase the data delivery latency [7], there are many promising wireless sensor network applications that are delay-tolerant and it is worthwhile to improve the performance of opportunistic data collection.

In the context of opportunistic data collection, the sensor data can be collected from a sensor node only after a mobile node approaches and they become aware of each other. Here, the event of the mobile node encountering a sensor node is referred to as a *contact*. To become aware of each other, beacons must be sent out by either mobile node or sensor node for *probing potential contacts*. As the movement of these mobile nodes is uncontrollable, *contact probing* becomes a challenging task for sensor nodes which need to be aggressively duty-cycled to achieve a long life. In this paper, the challenges faced by contact probing are studied systematically and the following contributions are made.

First, we investigate the ways that sensor nodes and mobile nodes carry out contact probing and propose a Sensor Node-Initiated Probing (SNIP) mechanism for improving data transfer capacity, the probed *contact capacity*, when the duty-cycle of a sensor node is fixed. SNIP is designed based on two reasonable assumptions, i.e. the radio of mobile nodes, which have relatively abundant energy via a rechargeable battery, can be always turned on [11] and the radio of sensor nodes designed for short-range communication (the Telos mote [12], etc.) consumes almost the same amount of energy in transmitting and receiving/listening modes. The basic principle of operation is that the sensor node initiates probing rather than a mobile node. Thus, a sensor node must broadcast a beacon immediately after its radio is turned on according to its duty-cycle. SNIP has been studied through analysis, extensive simulations and testbed experiments. The relationship between the percent of the probed contact capacity, sensor node's duty-cycle and the length of a contact is first modeled for SNIP and mobile node-initiated probing (MNIP) mechanisms from the literature. SNIP is then implemented in Contiki-OS [13] and extensive simulations are carried out using COOJA [14]. Both the analysis and simulation results indicate that SNIP outperforms MNIP mechanisms, and we quantify the impact of key parameters. Key observations are that, compared with MNIP mechanisms, with a sensor node duty-cycle that is lower than 1% the contact capacity probed by SNIP can be increased by a factor of 2–10; alternatively, for probing the same amount of contact capacity, the energy consumed by SNIP is much less than the energy consumed by MNIP mechanisms. SNIP is also implemented in Tiny-OS [15] and evaluated on a small testbed. The results confirm that SNIP does outperform MNIP mechanisms on the real hardware platforms. Hence, instead of mobile nodes, *sensor nodes should be responsible for broadcasting beacons*.

For our second contribution, since the intended applications of opportunistic data collection are delay-tolerant, a sensor node could have a lot of freedom when scheduling SNIP operations and we propose SNIP-Rush Hours (SNIP-RH) for improving the performance of contact probing through exploiting the temporal locality of human mobility identified in [10, 16]. More specifically, SNIP is mainly carried out when the sensor node believes that it is visited frequently and this node has enough sensor data to be uploaded in the next probed contact. The used duty-cycle is selected based on the contact length learned online. SNIP-RH is implemented in Contiki-OS and has been evaluated under a typical simulated roadside wireless sensor network scenario through both analysis and simulations. Based on the Nokia MDC Dataset that contains GPS readings of almost 200 participants collected over a period that exceeded one year [17, 18], SNIP-RH is also evaluated using real-world human mobility traces. Evaluation results indicate that SNIP-RH can significantly reduce the energy consumed for probing the necessary contacts to upload its sensor data, or significantly increase the probed contact capacity under a sensor node's energy budget for contact probing. Hence, *the temporal locality of human mobility should be exploited when carrying out contact probing for opportunistic data collection*.

This paper is organized as follows. Section 2 first investigates contact probing issues in opportunistic data collection

[1] IEEE 802.15.4 is the most widely used radio on sensor nodes and it starts to appear on smartphones. In Mobile World Congress 2012, TazTag released the first smartphone with both ZigBee (IEEE 802.15.4 radio, protocol stack, etc.) and Near Field Communication features (http://www.taztag.com/).

and formulates the problems to be solved. Section 3 then presents SNIP and its evaluation results in detail. SNIP-RH and its evaluation results are presented in Section 4. Finally, Section 5 discusses related work and Section 6 concludes.

2. PROBLEM FORMULATION

The reference network scenario of opportunistic data collection with smartphones is illustrated in Fig. 2. The mobile node's mobility is, from the point of view of the wireless sensor network, uncontrollable and cannot be predicted accurately by the sensor node. Hence, although a straight line with an arrow is shown in Fig. 2, we do not make any assumption about the movement trajectory and the speed. The radios of mobile node and sensor node could also have different communication ranges and different coverage shapes. In this paper, the event of the mobile node being in range of a sensor node is referred to as a contact. The length of a contact (α), i.e. the duration for which the mobile node and the sensor node stay within the communication range of each other, is the only parameter that characterizes a contact.

We note that multiple mobile nodes might move together and some sensor nodes might be deployed densely. In these cases, we can adopt some collision avoidance or contention resolution techniques [19] and allow a sensor node (mobile node) to choose one of these mobile nodes (sensor nodes) randomly or based on the radio signal strength, the movement speed, etc. For simplicity, these issues are omitted in this paper and we assume that at any time a sensor node can communicate with at most one mobile node and vice versa.

When carrying out contact probing, the radio of a sensor node is duty-cycled for achieving a long lifetime. More specifically, the radio is turned on for a fixed period (T_{on}) and alternately turned off for another fixed period (T_{off}). Hence, the duration of a cycle (T_c) is the sum of T_{on} and T_{off}. As for the duty-cycle of sensor node (d), it equals T_{on}/T_c.

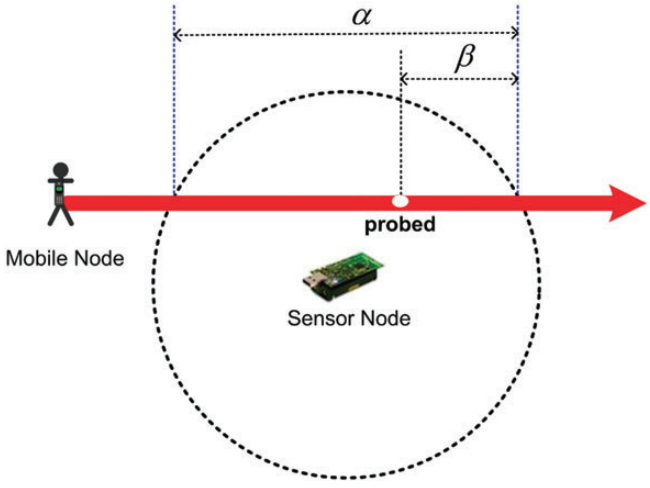

FIGURE 2. Contact probing reference model.

Under this scenario, the sensor data can be collected from a sensor node only after a mobile node approaches and they become aware of each other. The period that could be used for data collection starts immediately after they are aware of the presence of each other and its length (β) can be used to derive the amount of sensor data that could be collected in this contact. For a contact probing mechanism, it should be designed so that a contact can be successfully probed with high probability and the contact is probed as early as possible. More specifically, when a sensor node's duty-cycle is fixed, a contact probing mechanism should try to maximize $\Upsilon = \beta/\alpha$, the percent of contact capacity that is probed successfully for data collection.

For contact probing in opportunistic data collection, there are four processes in the system: the movement of a mobile node, the radio schedule of a mobile node, the radio schedule of a sensor node and the beacons periodically broadcasted by either mobile node or sensor node with an interval (T_b). To establish successful contact, a beacon must be sent out by either mobile node or sensor node when they are close to each other and their radios are both turned on. This can be difficult to achieve when the mobile node's movement is uncontrollable and the sensor node must be aggressively duty-cycled for reasons of longevity.

Since the mobility in opportunistic data collection is by definition uncontrollable, a contact probing mechanism is limited to controlling the broadcasting of beacons and the radio schedules of mobile node and sensor node. Considering that a mobile node could have relatively abundant energy via a rechargeable battery [11, 12], the radio of mobile node could be always turned on. Hence, a contact probing mechanism needs to answer the following key questions.

(1) To improve the probed contact capacity when the duty-cycle of a sensor node is fixed, who should be responsible for broadcasting beacons?
(2) To energy-efficiently probe the necessary contacts for uploading its sensor data, how should the sensor node select the duty-cycles to use?

In this paper, an SNIP mechanism is proposed to answer the first question. For the second question, SNIP-RH is proposed to exploit the temporal locality of human mobility. SNIP and SNIP-RH will be presented in the following sections and Table 1 lists the notations used in this paper.

3. SENSOR NODE-INITIATED PROBING

The energy consumed in transmitting messages can be a significant burden for wireless sensor nodes and thus it is logical to rely on resource-rich mobile nodes to broadcast beacons for the purpose of achieving synchronization and

TABLE 1. Notations.

	Comments
T_{on}	The period that a sensor node's radio is turned on
T_{off}	The period that a sensor node's radio is turned off
T_c	The cycle length of a duty-cycled sensor node. $T_c = T_{on} + T_{off}$
d	The duty-cycle of a sensor node. $d = T_{on}/T_c$
α	The length of a contact
β	The amount of time that could be used for data collection during a probed contact
Υ	The percent of probed contact capacity ($\Upsilon = \frac{\beta}{\alpha}$)
T_b	The interval between two consecutive beacons
T_p	The time needed for transmitting a beacon
∇	The threshold used by a sensor node to detect whether a mobile node has left
I'	The interval between two consecutive contacts
Γ	The epoch length of the repeated mobility pattern followed by mobile nodes
ζ	The contact capacity probed during an epoch
Φ	The contact probing overhead during an epoch. It is the total amount of time that the radio is turned on for contact probing
ρ	The energy cost per-unit of the probed contact capacity ($\rho = \Phi/\zeta$)

data transfer. However, these MNIP mechanisms face severe challenges in opportunistic data collection with smartphones. More specifically, since a sensor node must be duty-cycled, its radio schedule is unlikely to synchronize with the beacons emanating from a mobile node. In [20], it is proposed to set T_{on} of a sensor node according to T_b, the interval between two consecutive beacons from the mobile node. More formally, $T_{on} = T_b + T_p$, where T_p is the time needed for transmitting a beacon packet. The authors argue that a contact will be definitely detected if a sensor node's radio is turned on during the contact. However, T_b could be large in opportunistic data collection to avoid overburdening a mobile node and occupying the wireless channel, especially when there may be no sensor node listening. So T_{on} must be large too, and T_{off} will become huge to maintain a low duty-cycle. Consequently, with very high probability, a sensor node's radio will not be turned on during a contact whose length tends to be shorter than the huge T_{off} and the contact would thus be missed. Furthermore, in opportunistic data collection with smartphones, sensor nodes and mobile nodes belong to different authorities and it is hard to coordinate the values of T_b and T_{on}. In this paper, the proposal in [20] will be referred to as MNIP-JOINT, and the scheme with a fixed and short T_{on} will be referred to as MNIP-BASIC. Both of these mechanisms will be studied and compared with SNIP.

Due to the above shortcomings of MNIP mechanisms, SNIP, a novel sensor node-initiated probing mechanism, is proposed for improving the performance of contact probing in opportunistic data collection. In this section, the design choices of SNIP are first discussed and its details are presented. Through analysis, extensive simulations and testbed experiments, SNIP is then evaluated and compared with both MNIP-JOINT and MNIP-BASIC.

3.1. Design choices

Our key observation is that the low-power radio of the mainstream sensor node platforms consumes almost the same amount of energy in transmitting and receiving/listening modes. For example, the IEEE 802.15.4-Compliant CC2420 radio of TELOSB node consumes 35 mW when transmitting at its default power level (0 dBm) and it consumes 38 mW in receiving mode [12]. The current of the nRF24AP1 radio for an ANT wireless sensor network is 22 mA in receiving mode and the current in transmitting mode is 13.5 mA [21]. Hence, with such a platform, it is effectively free, in terms of energy usage, for a sensor node to broadcast a beacon when its radio is already turned on.

Another observation is that a mobile node could be equipped with a relatively abundant and rechargeable power supply and its radio used for opportunistic data collection can be always turned on. This is true for smartphones, on which opportunistic data collection would be treated as a lower priority task. For example, it is claimed that the talk time of Google Nexus One smartphone is 7 h and the smartphone consumes about 746.8 mW during a voice call [11]. Considering the energy consumed by the CC2420 radio on a TELOSB node, even if IEEE 802.15.4 radio consumes more energy on the smartphone platform, the smartphone's battery still could last a few days when the IEEE 802.15.4 radio is installed and is always turned on. Furthermore, without undermining the assumption that the smartphone's IEEE 802.15.4 radio is always turned on for contact probing, there are a lot of opportunities to reduce the smartphone's energy consumption based on history and/or context information. For example, a smartphone can deduce whether it is moving based on accelerometer readings [22, 23]. When the smartphone is static and it has collected all data of the nearby sensor nodes, its radio for data collection can be turned off for saving energy. Considering that smartphone users are usually static [10], this scheme can significantly reduce the energy consumed by the smartphone's low-power radio for contact probing.

Based on the above observations, the radio of a mobile node is assumed to be always turned on in SNIP. In contrast with MNIP mechanisms, a sensor node in SNIP is required to broadcast a beacon immediately after its radio is turned on. Since the radio of a mobile node is always turned on, if the sensor node broadcasts a beacon when they are close to each other, this contact will be definitely probed successfully,

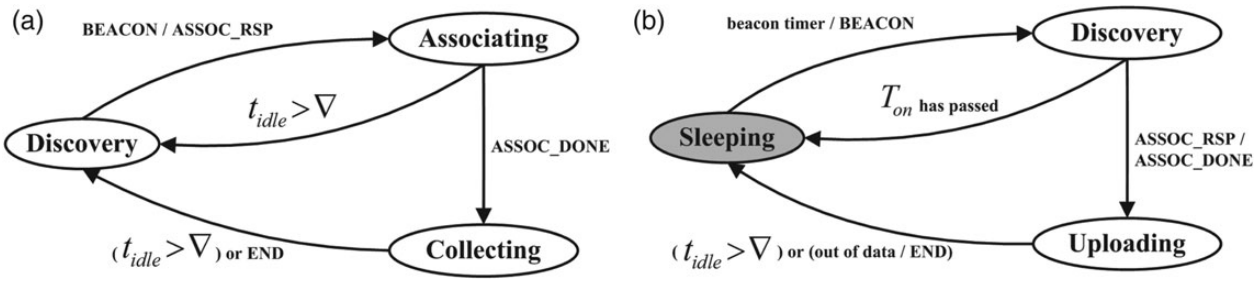

FIGURE 3. State transition diagram of SNIP. (a) Mobile node. (b) Sensor node.

assuming of course that the beacon is not lost or corrupted due to contention, which is unlikely in sparse deployments and short-range transmissions. Considering that a sensor node can turn on/off its radio relatively quickly, T_{on} can be set to a small value and a sensor node can carry out contact probing frequently. Hence, a contact will be probed successfully with high probability, and the probed contact capacity will be increased significantly.

3.2. Details of SNIP

Figure 3a shows the state transition diagram of a mobile node in SNIP. In SNIP, a mobile node moves around in an uncontrolled manner and its radio is always turned on so that it can be discovered. After receiving a BEACON from a sensor node, a mobile node will send back ASSOC_RSP and enter into Associating state. After receiving ASSOC_DONE from the sensor node, the association is complete. The mobile node will enter into Collecting state and start to collect data from the sensor node. In Collecting state, the contact may be terminated by the sensor node through sending END to the mobile node. In both Associating and Collecting states, the mobile node also keeps monitoring whether it has moved away from the sensor node. When it finds that t_{idle} (the time that the channel is idle) is larger than a constant (∇), the mobile node returns back to Discovery state and is ready to be discovered again. We see that ∇ should be long enough to avoid that the mobile node terminates data collection unnecessarily.

Figure 3b shows the state transition diagram of the sensor node in SNIP. When a sensor node's beacon/duty-cycle timer expires, it will turn on its radio, send out a BEACON and enter into Discovery state. If it does not receive an ASSOC_RSP within T_{on}, the sensor node will turn off its radio, return back to the Sleeping state and start its beacon/duty-cycle timer for the next probing; T_{on} should be long enough for sending a BEACON and receiving a ASSOC_RSP. If ASSOC_RSP is received in Discovery state, it will send back ASSOC_DONE, enter into Uploading state and start to transfer data to the associated mobile node. The simple Stop-and-Wait protocol is used for flow control, a retransmission timer is used for reliable data transmission and multiple sensor reports are concatenated into one packet for reducing header overhead. If all data have been uploaded, the sensor node will send END to the mobile node for terminating this contact. In Uploading state, the sensor node also keeps monitoring whether the mobile node has moved away. When it finds that t_{idle} is larger than ∇, the sensor node will turn off its radio, return back to Sleeping state and start its beacon/duty-cycle timer for the next probing.

3.3. Modeling

In this subsection, we will model SNIP with a focus on the relationship between Υ (percent of the probed contact capacity), d (the duty-cycle used by a sensor node during contact probing) and α (the length of a contact). More specifically, β is modeled and Υ can be deduced immediately, i.e. $\Upsilon(d, \alpha) = \beta/\alpha$. For comparison, MNIP-BASIC and MNIP-JOINT are also modeled in the same manner.

3.3.1. SNIP

Figure 4 shows the three processes in SNIP: the occurrence of a contact, the sensor node radio which also incorporates the beacon emanating process and the mobile node radio.

In SNIP, the sensor node will broadcast a BEACON when its radio is turned on. Hence, T_{b}, the interval between two consecutive beacons, equals T_{c}. We see that x is the difference between the time that the last beacon is broadcasted and the time that a contact occurs (i.e. a mobile node moves into the communication range of a sensor node). Since the mobility of smartphone users is uncontrollable, even though their mobility may follow some patterns over a large time scale [10] and the temporal locality is exploited in Section 4, during a short

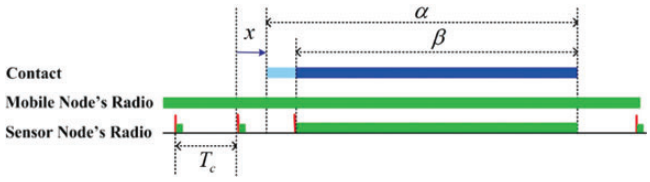

FIGURE 4. Time line of SNIP, a sensor node-initiated probing protocol.

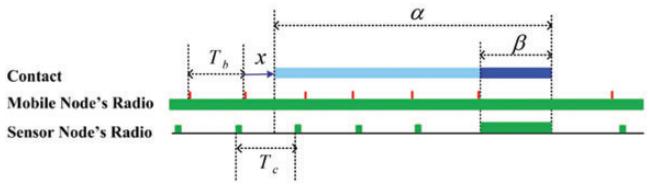

FIGURE 5. Time line of MNIP-BASIC.

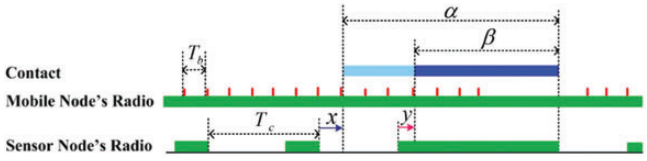

FIGURE 6. Time line of MNIP-JOINT.

period of T_c, a contact can occur at any time with the same probability. Since T_c is much larger than the time needed for transmitting a BEACON, we do not consider the case where a mobile node arrives during the transmission of a BEACON. Hence, we can assume that x is uniformly distributed between 0 and T_c. The expectation of β can then be modeled as $\beta = (1/T_c)\int_0^{T_c} \beta(x)\,dx$. As shown in Fig. 4, it is obvious that

$$\beta(x) = \{(x+\alpha) - T_c\}^+ \tag{1}$$

Here, $\{\cdot\}^+$ is defined as $\max(0, \cdot)$.

3.3.2. *MNIP-BASIC*

In MNIP-BASIC, the mobile node periodically broadcasts beacons and the value of T_b is selected based on its own situation. We see that T_{on} is a small and fixed value selected by the sensor node.

As illustrated in Fig. 5, x is also the difference between the time that the last beacon is broadcasted and the time that a contact occurs. Hence, the number of beacons transmitted during a contact, can be modeled as follows:

$$\aleph_b(x) = \left\lfloor \frac{\{\alpha - (T_b - x)\}^+}{T_b} \right\rfloor \tag{2}$$

For simplicity, we assume that the interval between two consecutive beacons fluctuates a little around T_b so that the synchronization between the beacon process of the mobile node and the radio schedule of the sensor node can be avoided. For each beacon transmitted during the contact, the probability that it is successfully received by the sensor node can then be modeled simply as $p_s = (T_{on} - T_p)/T_c$. Here, T_p is the time needed to transmit a beacon packet. Due to the uncontrollable mobility of smartphone users, we also assume that x is uniformly distributed between 0 and T_b. Similar to Section 3.3.1, the expectation of β is modeled as $\beta = (1/T_b)\int_0^{T_b} \beta(x)\,dx$. However, the value of $\beta(x)$ is affected by the beacon that is received by sensor node successfully, and $\beta(x)$ can be modeled as follows:

$$\beta(x) = \sum_{i=0}^{\aleph_b(x)-1} [(1-p_s)^i * p_s * (\alpha - (T_b - x) - i * T_b)] \tag{3}$$

3.3.3. *MNIP-JOINT*

Unlike MNIP-BASIC in which T_{on} of the sensor node is set to a fixed small value, MNIP-JOINT sets T_{on} of the sensor node according to T_b selected by the mobile node. More formally, $T_{on} = T_b + T_p$.

As illustrated in Fig. 6, x is the difference between the time that sensor node turns off its radio and the time that a contact occurs. By y we denote the difference between the time that the sensor node turns on its radio and the time that the mobile node transmits a beacon after that. Since the mobility of smartphone users is uncontrollable, we assume that x is uniformly distributed between 0 and T_c, and y is uniformly distributed between 0 and T_{on}. Consequently, the expectation of β can be modeled as $\beta = (1/T_c)\int_0^{T_c} \beta(x)\,dx$. Note that $\beta(x)$ is affected by the value of y and it can be modeled as $\beta(x) = (1/T_{on})\int_0^{T_{on}} \beta(x,y)\,dy$. As shown in Fig. 6, in the case where $x < (T_{off} + y)$,

$$\beta(x,y) = \{(x+\alpha) - (T_{off} + y)\}^+ \tag{4}$$

In the case where $x > (T_{off} + y)$, the contact still may be probed successfully when the sensor node turns on its radio again. Hence,

$$\beta(x,y) = \{\alpha - (T_c - x) - (T_{off} + T_{on}/2)\}^+ \tag{5}$$

3.4. Numerical results

To study SNIP and compare it with MNIP-BASIC and MNIP-JOINT, the curves between Υ and d are calculated numerically based on the above models for several typical values of α: 2, 5, 10 and 30 s. These values represent the time needed by a car on a freeway, a car in the city, cycling people and walking people to pass through a distance of 50 m, which is selected according to the communication range of current sensor node platforms. The duty-cycle of the sensor node varies from 0.001 to 0.2.

We set T_{on} to 20 ms in both SNIP and MNIP-BASIC. This value should be large enough for the current sensor node platform to send a BEACON and receive a ASSOC_RSP. In MNIP-JOINT, T_{on} is set to the sum of T_b and T_p. According to the current sensor node platform, T_p is set to 10 ms. In both MNIP-BASIC and MNIP-JOINT, the evaluated values of T_b are 100 and 500 ms. The smaller values of T_b are not chosen because smartphones will be overburdened and the wireless channel will be occupied by beacons even when the sensor node does not exist. We set ∇, which is used to detect whether the mobile node has moved away, to 50 ms and this value is long enough to (re)transmit a packet a few times through the

TABLE 2. Parameter values.

	SNIP	MNIP-BASIC	MNIP-JOINT
T_{on}	20 ms	20 ms	$T_b + T_p$
T_b	T_c	100, 500 ms	100, 500 ms
T_p	N/A	10 ms	10 ms
∇	50 ms	50 ms	50 ms

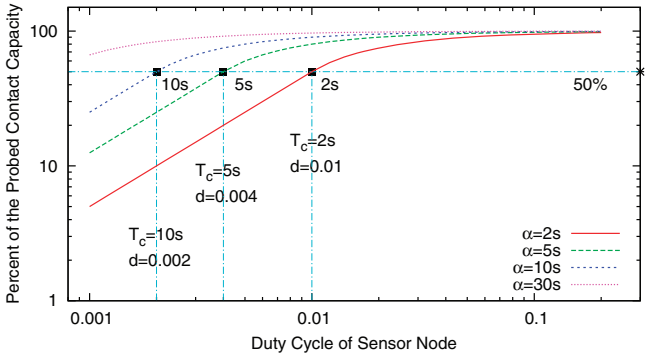

FIGURE 7. Numerical results of SNIP.

wireless channel. Table 2 summarizes the parameter values of these contact probing mechanisms.

3.4.1. *SNIP analysis*

Figure 7 plots the numerical results of SNIP. The X-axis is the duty-cycle used by a sensor node. Note that the energy consumption of a sensor node is proportional to the duty-cycle and the duty-cycle of a sensor node can be used to depict the energy consumed by contact probing. The Y-axis is the percent of the probed contact capacity, which determines the amount of probed contact capacity. Due to the wide range of Υ and the evaluated duty-cycles, both axes use a logarithmic scale.

Figure 7 indicates that Υ increases with d and α significantly affects the curve. It is obvious that Υ should increase with d. When d is fixed, Υ also increases with α. The reason is that when α is larger, the mobile node stays in the communication range of the sensor node for a longer time and it becomes much easier for them to discover each other. Figure 7 also indicates that, when $T_c \geq \alpha$, Υ is linearly related with d. In fact, the following closed-form equations can be deduced through modeling the following two cases separately.

When $T_c \geq \alpha$, the expected probability that a contact is probed successfully could be modeled as $E[p_s] = \alpha/T_c$, and the expected period that could be used for data collection can be modeled as $E[\beta_s] = \alpha/2$. Thus,

$$\beta = E[p_s] * E[\beta_s] = \left(\frac{\alpha}{T_c}\right) * \left(\frac{\alpha}{2}\right) = \frac{\alpha^2}{2 * T_{on}} * d \qquad (6)$$

When $T_c < \alpha$, $E[p_s] = 1$ and $E[\beta_s] = \alpha - T_c/2$. Thus,

$$\beta = E[p_s] * E[\beta_s] = 1 * \left(\alpha - \frac{T_c}{2}\right) = \alpha - \frac{T_{on}}{2 * d} \qquad (7)$$

Consequently,

$$\Upsilon = \frac{\beta}{\alpha} = \begin{cases} \dfrac{\alpha}{2 * T_{on}} * d, & T_c \geq \alpha \\ 1 - \dfrac{T_{on}}{2 * d * \alpha}, & T_c < \alpha \end{cases} \qquad (8)$$

3.4.2. *Comparison of SNIP, MNIP-BASIC and MNIP-JOINT*

To compare with MNIP-BASIC and MNIP-JOINT, for each value of α, Fig. 8 plots the curves of these models together. It shows that SNIP outperforms the MNIP mechanisms, especially when the duty-cycle is low.

3.5. Simulation results

To evaluate SNIP in more realistic environments, SNIP is implemented in Contiki-OS [13] and extensive simulations are carried out in COOJA [14], which incorporates a machine code instruction level emulator of the TELOSB node. For comparison, MNIP-BASIC and MNIP-JOINT are also implemented in Contiki-OS and simulated in COOJA. When implementing these contact probing mechanisms in Contiki-OS, the same parameter values given in Table 2 are adopted. In the following simulations, we let a mobile node visit a sensor node repeatedly and the sensor node always has sensor data to be uploaded.

For validating the SNIP model, simulations are designed based on the above numerical study. The evaluated duty-cycles of the sensor node are 0.001, 0.002, 0.004, 0.01, 0.02, 0.04, 0.1 and 0.2. The evaluated values of α (contact length) are 2, 5, 10 and 30 s. When generating mobility traces used in simulations, both a normal distribution with small deviation (a 10th of the mean) and an exponential distribution are studied for α. For the interval between two consecutive visits (I'), the evaluated values are 100, 200, 500 and 1000 s, and we also consider both a normal distribution and an exponential distribution. However, as indicated in the above analysis, although I' affects the amount of probed contact capacity, Υ has no relationship with I'. Our simulation results do confirm this observation. Hence, in this paper, we will just plot the results when I' is 200 s and follows a normal distribution. For the accuracy of simulation results, each experiment is run for 100 h, which should be long enough since a sensor node is visited 1800 times when $I' = 200$ s.

3.5.1. *Validation of the SNIP model*

To validate our SNIP model in Section 3.3, for each distribution followed by α, the simulation results and numerical results of SNIP with different values of α are plotted together. Figure 9a

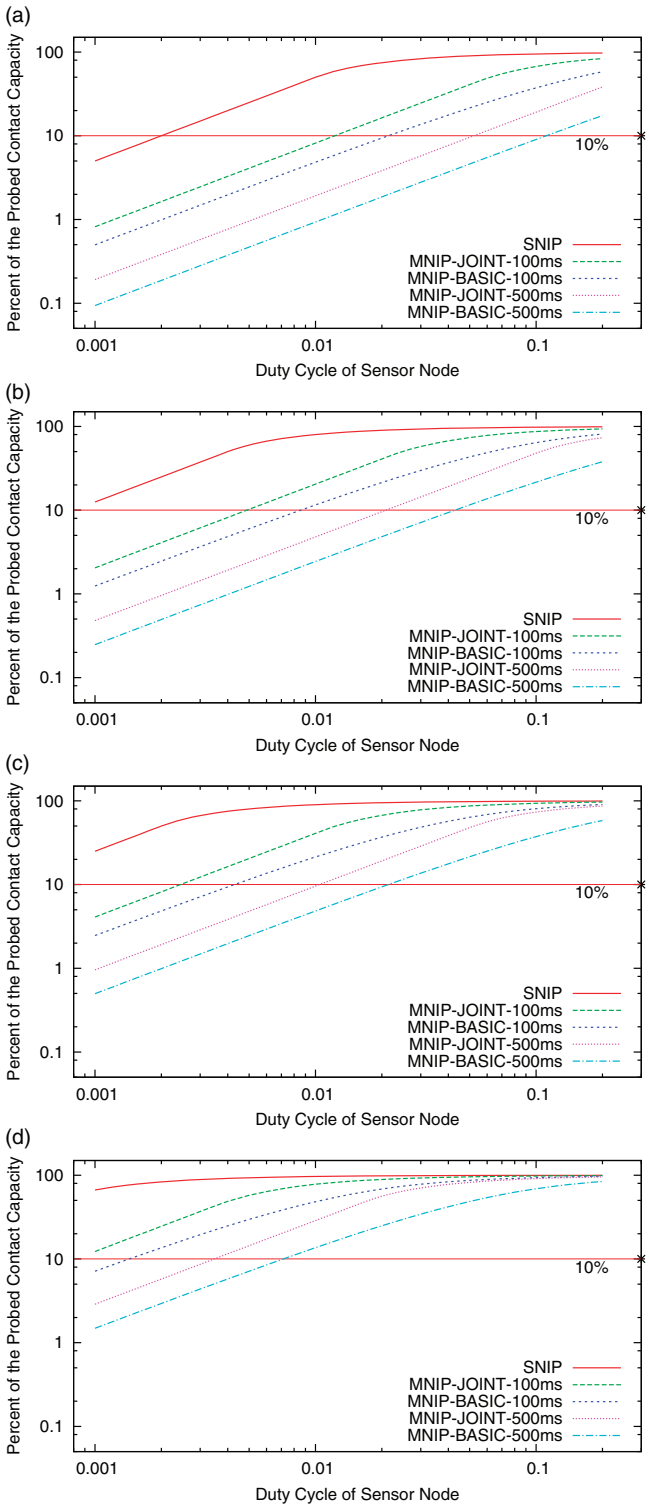

FIGURE 8. Comparison of SNIP and MNIP. (**a**) $\alpha = 2$ s, (**b**) $\alpha = 5$ s, (**c**) $\alpha = 10$ s and (**d**) $\alpha = 30$ s.

and b shows the plots when α follows the normal distribution and the exponential distribution, respectively.

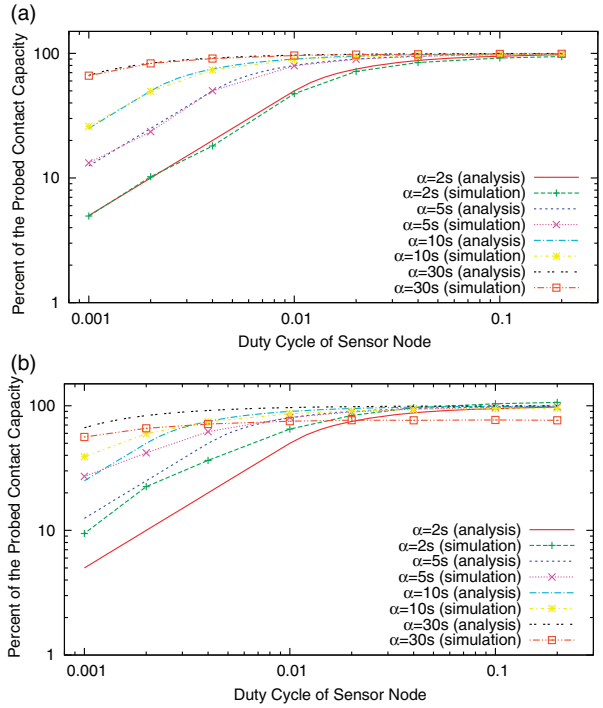

FIGURE 9. Validation of the SNIP model. (**a**) Contact length: normal distribution. (**b**) Contact length: exponential distribution.

Figure 9a indicates that our SNIP model is very accurate when the contact length follows the normal distribution. It means that our model does capture the fundamentals of SNIP. However, as shown in Fig. 9b, when contact length follows the exponential distribution, there are some differences between our model and the simulation results. The simulation results are obviously better than the numbers calculated based on our model, especially when the duty-cycle of the sensor node is low. The reason is that the variance of the contact length is much larger when it follows the exponential distribution. For modeling, it becomes insufficient to consider only the mean of the contact length. As shown in Equation (9), the distribution of the contact length should be modeled explicitly.

$$\overline{\Upsilon} = \frac{1}{\overline{\alpha}} * \int_0^\infty \alpha * \Upsilon(d, \alpha) * P(\alpha)\, \mathrm{d}x \tag{9}$$

Here, $P(\alpha)$ is the probability that the length of a contact is α. As illustrated in Fig. 7, in the case where the duty-cycle of the sensor node is low, Υ of the contacts that are longer than the mean can be much larger than $\Upsilon(d, \overline{\alpha})$. Since these long contacts tend to be a significant part of the overall contact capacity, the simulation results will be better than the output of the model based on only the mean of the contact length.

3.5.2. Comparison of SNIP, MNIP-BASIC and MNIP-JOINT
To compare SNIP with MNIP-BASIC and MNIP-JOINT, for each combination of the evaluated value and the distribution of

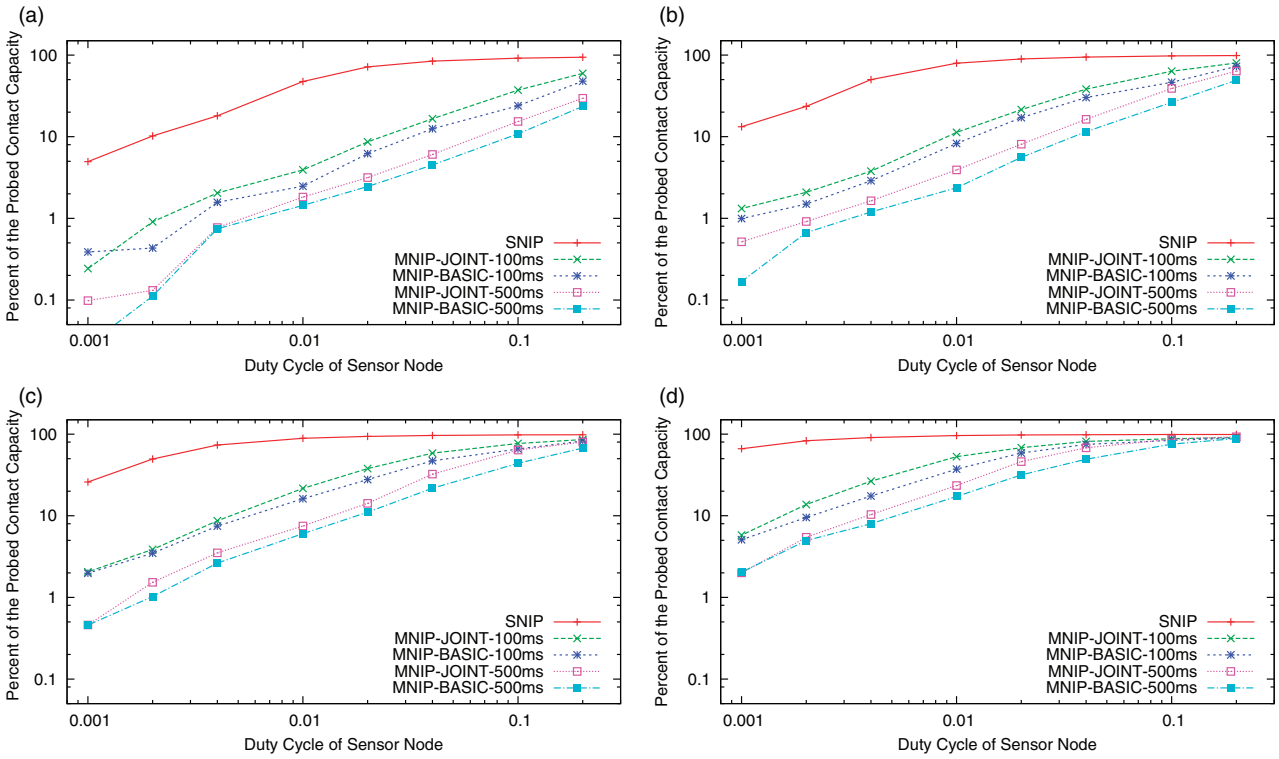

FIGURE 10. Simulation results of SNIP, MNIP-BASIC and MNIP-JOINT (normal). (**a**) $\alpha = 2$ s, (**b**) $\alpha = 5$ s, (**c**) $\alpha = 10$ s and (d) $\alpha = 30$ s.

α, simulation results of SNIP, MNIP-BASIC and MNIP-JOINT are plotted together.

Figure 10 shows the results when α follows the normal distribution and Fig. 11 shows the results when α follows the exponential distribution. These plots indicate that SNIP performs much better than MNIP mechanisms in all cases, especially when the duty-cycle is low. When the duty-cycle is lower than 1%, compared with MNIP-JOINT with a high probing frequency (T_b=100 ms), SNIP can improve the performance by a factor of 2–10.

3.6. Testbed evaluations

To verify the performance of SNIP, it is also evaluated on real hardware. Instead of using the code developed on Contiki-OS and used in simulations with COOJA, SNIP is implemented on Tiny-OS [15] to demonstrate its simplicity and portability. For comparison, MNIP-BASIC and MNIP-JOINT are also implemented in Tiny-OS and T_b is set to 100 ms in the following experiments.

A small testbed is set up at five far-apart positions in our department building. At each position, three USN MTM-CM5000-MSP nodes (a clone of TELOSB) from MAXFOR Technology Inc. are deployed for running SNIP, MNIP-BASIC and MNIP-JOINT, respectively. Note that three different channels are used to avoid the interference among these nodes. USN MTM-CM5000-MSP nodes are also used as the mobile nodes. More specifically, when carrying out the experiments, a student carried three nodes and moved around in the building to collect data from sensor nodes that were running SNIP, MNIP-BASIC or MNIP-JOINT. Thus, these three contact probing mechanisms are evaluated simultaneously in the same environment with the same user mobility.

In the experiments, we still assume that a sensor node always has sensor data to be collected. To get meaningful results in a short time, the student will visit the five positions periodically (once per 2 h) and stay at each position for a few seconds (∼20 s). We have evaluated the results when the duty-cycle of the sensor node equals 0.001, 0.002, 0.0033, 0.005 and 0.01. For each duty-cycle, the experiment will last for 3 days.

Figure 12 shows the experiment results. Since we do not know the exact amount of the overall contact capacity, instead of the percent of probed contact capacity, we plot the network throughput, i.e. the total number of bytes collected from all sensor nodes. Due to the wide range of the collected bytes, for each evaluated duty-cycle, we plot the normalized network throughput (normalized to MNIP-BASIC) together. Figure 12 indicates that SNIP does perform much better than MNIP mechanisms on the real hardware platforms. SNIP can improve the amount of collected data by 150–370%.

All the above analysis, simulation and experiment results indicate that SNIP significantly outperforms MNIP

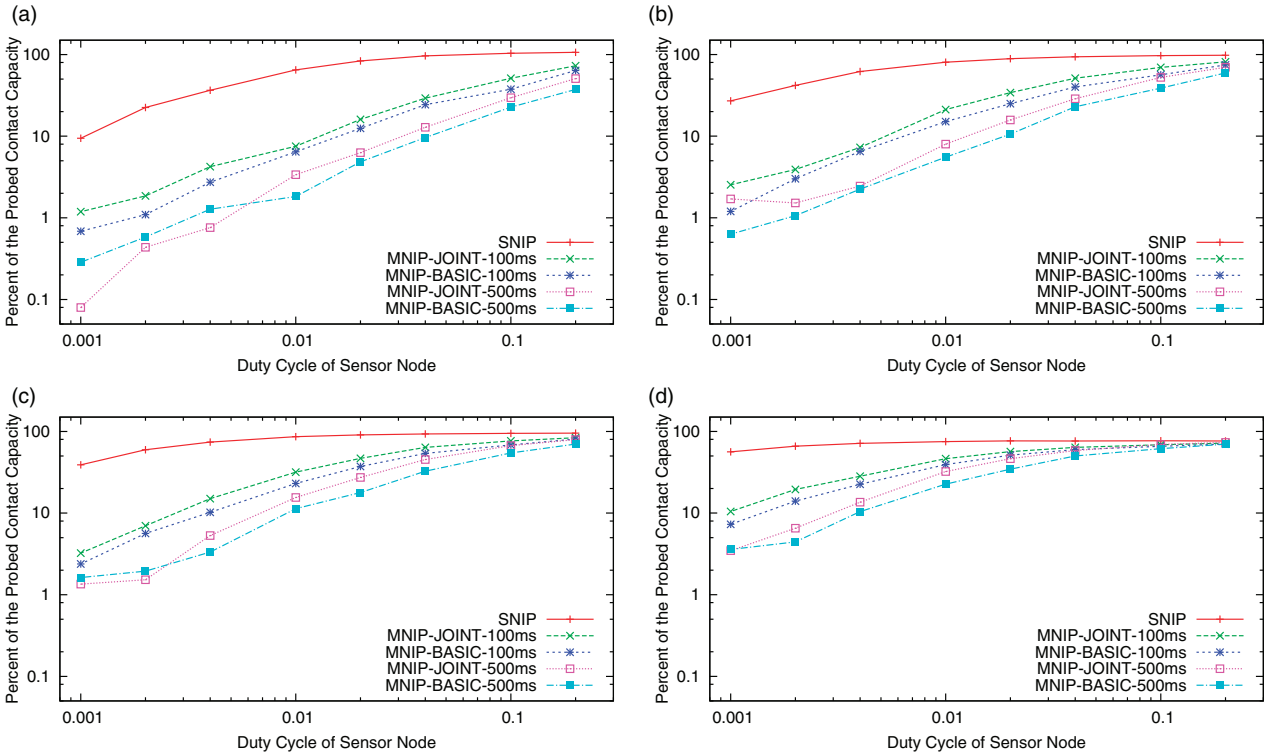

FIGURE 11. Simulation results of SNIP, MNIP-BASIC and MNIP-JOINT (exponential). (**a**) $\alpha = 2$ s, (**b**) $\alpha = 5$ s, (**c**) $\alpha = 10$ s and (**d**) $\alpha = 30$ s.

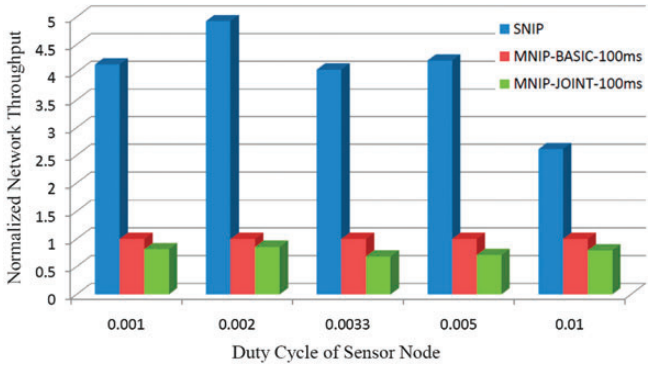

FIGURE 12. Testbed evaluation results of SNIP.

mechanisms, especially when the duty-cycle of a sensor node is low. For opportunistic data collection in which sensor nodes must be deeply duty-cycled, SNIP still can be much more energy-efficient even if the energy consumed by the radio of the sensor node in transmitting mode is more than that consumed in receiving/listening mode. Hence, SNIP should also be a promising solution when sensor nodes and smartphones communicate through Bluetooth that consumes more energy in transmitting mode.[2]

[2]Bluetooth is distributed with almost all current smartphones and it is also adopted by many sensor nodes, such as IMote and BTnode.

4. SNIP-RUSH HOURS

Through evaluating SNIP extensively, we have answered the first question in Section 2. Instead of mobile nodes, the duty-cycled sensor nodes should broadcast the beacons and SNIP should be used to probe contacts in opportunistic data collection. To answer the second one, we will study how to schedule the SNIP operations of a sensor node, i.e. how to select the duty-cycles used at different times. In this section, we first discuss the motivations for exploiting the temporal locality of human mobility. The duty-cycle selection problem is then modeled as an optimization problem. After that, SNIP-RH, a practical duty-cycle selection mechanism, is described in detail. Its analysis and simulation results are then presented and discussed at the end of this section. Note that the smartphone users' mobility traces from Nokia Mobile Data Challenge [17, 18] have been used to validate SNIP-RH under more realistic environments.

4.1. Motivation

The straightforward way is to probe contacts in *all time* with a fixed duty-cycle, which is selected so that the amount of probed contact capacity is just enough to upload its sensor data. This mechanism will be referred to as SNIP-AT. However, considering that the intended applications are delay-tolerant, a sensor node could have a lot of freedom when scheduling

SNIP operations and there should be further opportunities for improving the performance of contact probing. SNIP-RH is thus motivated by the following two further observations.

First, the mobility of mobile nodes (i.e. smartphone users) normally follows some repeated mobility patterns. It has been shown that human trajectories have a high degree of temporal and spatial regularity and their mobility follows simple reproducible patterns [16, 24]. The temporal distribution of human mobility also shows high variance [25]. Hence, the contacts between a sensor node and mobile nodes tend to arrive unevenly across time and *rush hours* will exist in the temporal distribution of these contacts. In [26], the authors studied the temporal distribution of user travel demand at the Midpoint Bridge, FL, USA. They found that rush hours do exist and do not disappear even after a time-variable pricing scheme is adopted by the toll bridge for spreading the travel demand. The possible reason is that people must live according to the same timetable agreed by the whole society and it cannot be changed easily.

Through analyzing the smartphone users' mobility traces from Nokia Mobile Data Challenge that contain GPS readings of almost 200 participants in the course of 1 + year [17, 18], we have validated these findings in a more appropriate spatial granularity for opportunistic data collection [10]. Hence, temporal locality does and will exist in the mobility of smartphone users. Due to the repeated pattern followed by their mobility, it also becomes possible for a sensor node to learn and/or predict rush hours, during which mobile nodes visit more frequently. Furthermore, we find that, for different locations, smartphone users may visit them with different temporal patterns. Figure 13 illustrates that two locations in Lausanne, Switzerland have totally different rush hours. Thus, instead of one probing strategy for all sensor nodes, each node should learn and adapt to the temporal pattern that smartphone users visit that node.

Second, according to the results in Section 3, Υ (the percent of the probed contact capacity) is independent of the value and the distribution of I' (the time between two consecutive contacts). However, I' does affect the overall contact capacity. Consequently, when probing contacts with the same duty-cycle, the amount of probed contact capacity during rush hours (the periods that contacts arrive much more frequently) will be much higher than that probed during other periods of the same length. Alternatively, in rush hours, the same amount of contact capacity can be probed with a much lower energy consumption. Hence, temporal locality of the smartphone users' mobility should be exploited when probing the contacts. More specifically, when it is possible, contact probing should be carried out mainly in rush hours for saving energy.

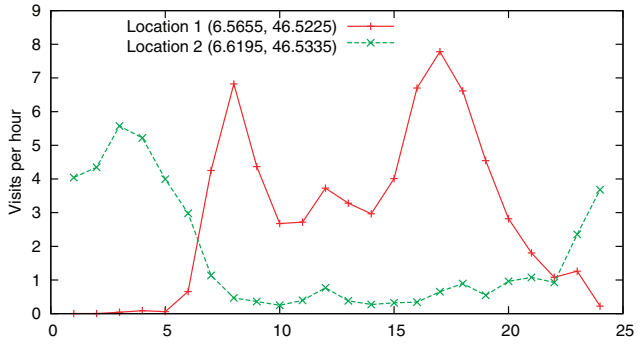

FIGURE 13. Temporal distributions of smartphone GPS readings for two locations in Lausanne.

In summary, temporal locality (i.e. rush hours) exists widely in the mobility of smartphone users. It should and could be exploited by a sensor node when scheduling SNIP operations.

4.2. Optimization-based SNIP scheduling

In this subsection, scheduling SNIP operations to exploit the temporal locality of human mobility will be modeled as an optimization problem. More specifically, we assume that a sensor node has a target of the probed contact capacity (ζ_{target}) which is just enough to upload the sensor data generated for satisfying the application requirements. We also assume that a sensor node has a budget for the energy consumed by contact probing ($\Phi_{\max}$) so that it can assure at least a minimum lifetime. When it is possible to probe the necessary contacts under the energy budget, a sensor node will try to minimize the energy consumption for extending its life. Otherwise, a sensor node will maximize the probed contact capacity under the energy budget and adjust its data rate accordingly.

We use Γ is used here to denote the epoch length of mobile nodes' repeated mobility pattern and an epoch is divided into n time-slots with the following length, $t_1, t_2, \ldots, t_n$. We assume that the contact arrival process of each time-slot (both contact arrival frequency and contact length distribution) can be learned accurately. Based on the learned contact arrival process and the closed-form equations of the SNIP model (Equations (8) and (9)), we can deduce $\zeta_i(d_i) = t_i/I'_i * \alpha_i * \Upsilon_i(d_i, \alpha_i)$, which is the amount of contact capacity probed during time-slot i when SNIP is carried out with a duty-cycle d_i. With a scheduling plan $(d_1, d_2, \ldots, d_n)$, the total amount of probed contact capacity is $\zeta = \sum_{i=1}^{i=n} \zeta_i(d_i)$ and the energy consumed for contact probing is $\Phi = \sum_{i=1}^{i=n} t_i * d_i$. Hence, the task of scheduling SNIP operations becomes a duty-cycle selection problem, in which a sensor node need to decide the value of d_i for each time-slot to optimize the performance of contact probing.

The duty-cycle selection problem can then be solved through the following two steps. In the first step, a sensor node tries to maximize ζ under the constraints that $\Phi \leq \Phi_{\max}$ and $0 \leq d_i \leq 1$ $(i = 1, 2, \ldots, n)$. If the maximal ζ is less than ζ_{target}, the sensor node has obtained the optimal scheduling plan and it should reduce its data rate accordingly. Otherwise, the second step will be executed. In the second step, the sensor node

will try to minimize Φ under the constraints that $\zeta \geq \zeta_{\text{target}}$ and $0 \leq d_i \leq 1$ $(i = 1, 2, \ldots, n)$. Hence, the life of the sensor node can be maximized. This two-step optimization-based scheduling mechanism will be referred to as SNIP-OPT.

Although SNIP-OPT can produce the optimal scheduling plan, it may not be applicable since SNIP-OPT assumes that a sensor node knows the exact contact arrival process for each time-slot. As illustrated in Fig. 13, for different locations, the visits of smartphone users follow different temporal distributions. Thus, it would be hard for engineers to get this information for each sensor node. It is also difficult to enable sensor nodes to learn the information and execute SNIP-OPT autonomously. First, the CPU of the sensor node may not be powerful enough to solve the optimization problem. Note that the percent of probed contact capacity is not linearly related with the duty-cycle when the used duty-cycle is higher than a threshold. SNIP-OPT fundamentally includes two non-linear programming problems. Second, considering the large number of parameters to be estimated (contact arrival frequency and contact length distribution for each time-slot) and the low duty-cycle that must be used for longevity, it is very challenging for a sensor node to learn the information required by SNIP-OPT. In this paper, MATLAB is used to solve the optimization problem and the results are used to compare with SNIP-RH, a practical mechanism to be presented below.

4.3. Details of SNIP-RH

Although it is hard to know the exact contact arrival process of each time-slot, it should be easy to identify the time-slots expected to have high contact capacity. Based on this observation, SNIP-RH is designed to exploit the temporal locality of human mobility. The main principle of SNIP-RH is that SNIP is activated only during rush hours, the periods in which contacts are expected to arrive more frequently. Its details will be presented and discussed below.

4.3.1. Rush hours

An epoch of mobile nodes' repeated mobility pattern is divided into n time-slots with the same length and each time-slot is marked as '1' or '0'. '1' indicates that a time-slot is in a rush hour. Engineers are responsible for determining n and Γ based on the deployment environment; Γ should be equal to the epoch length of mobile nodes' repeated mobility pattern. On should select n based on the mobility pattern and the available resources of a sensor node. With a larger n, rush hours can be specified more accurately, but it takes more effort to identify rush hours among these time-slots.

Considering that the smartphones are used as mobile nodes and human mobility follows the diurnal pattern, Γ can be set to 24 h. One can set n to 24 so that the length of each time-slot is exactly 1 h.

4.3.2. SNIP scheduling

In SNIP-RH, we assume that the CPU of a sensor node wakes up periodically to decide whether to carry out contact probing. The sensor node will activate SNIP only when all of the following three conditions are satisfied.

(1) To exploit temporal locality of human mobility, the current time-slot must be marked as '1'.
(2) The sensor node has enough sensor data to be uploaded in the next probed contact. Hence, the probed contact capacity will not be wasted and the energy consumption for contact probing can be reduced. The threshold for available sensor data can be set according to the amount of sensor data uploaded in previous probed contacts.
(3) The energy that had been consumed for contact probing in the current epoch should be less than the sensor node's energy budget for contact probing.

Hence, a sensor node needs to maintain the energy that it has consumed for contact probing in the current epoch. It also needs to keep updating the average amount of sensor data uploaded during a probed contact. To filter out the noise in the amount of sensor data uploaded during a probed contact, an exponentially weighted moving average (EWMA) scheme is used and a small weight is assigned to the new sample.

4.3.3. The used duty-cycle

The mean of the contact length (α) is also learned for deciding the value of d_{rh} (the duty-cycle used by SNIP when it is activated). It is deduced based on the length of the probed contacts and the value of the used duty-cycle. To filter out the noise of contact length, an EWMA scheme is also used here and a small weight is assigned to the new sample. In SNIP-RH, d_{rh} is set to T_{on}/α and this design is made based on the following observation.

According to the SNIP model, when $d \leq T_{\text{on}}/\alpha$, Υ increases linearly with the increase of d. When d is increased further, Υ increases much more slowly. We use ρ here to denote the energy cost for per-unit of the probed contact capacity and equals Φ/ζ. By Φ we denote the energy consumed for contact probing and it is linearly related with d; ζ is the amount of probed contact capacity and it is linearly related with Υ. Consequently, when $d \leq T_{\text{on}}/\alpha$, ρ does not change with d and it equals the smallest value.

Hence, it is desirable if d_{rh} is no larger than T_{on}/α. Through setting d_{rh} to T_{on}/α, a sensor node can maximize the contact capacity probed during rush hours with the smallest ρ.[3] The evaluation results of SNIP (as shown in Figs 7, 9a and b) also indicate that ρ does not increase abruptly when d_{rh} is slightly

[3] As shown in Fig. 9, when α follows the exponential distribution, Υ is not linearly related with d even if $T_{\text{c}} \geq \alpha$, i.e. $d \leq T_{\text{on}}/\alpha$. However, we still can observe the obvious slope change at the point where $T_{\text{c}} = \alpha$. Hence, $d = T_{\text{on}}/\alpha$ should be a good selection for SNIP-RH even when α varies a lot.

larger than T_{on}/α. Hence, SNIP-RH is not sensitive to the estimation accuracy of α and it should work well in practice.

4.4. Evaluation results: roadside scenario

To evaluate SNIP-RH and compare it with both SNIP-AT and SNIP-OPT, these mechanisms are studied under the following scenario of a simplified roadside wireless sensor network: Γ is 24 h, n is set to 24 and the rush hours are 7:00 to 9:00 and 17:00 to 19:00. I' is 300 s in rush hours and it equals 1800 s at other times. All contacts have the same length, i.e. α equals 2 s.

To study how well these mechanisms perform under different situations, Φ_{max} is set to $\Gamma/1000$ and $\Gamma/100$ and ζ_{target} varies from 16 to 56 s.

4.4.1. *Numerical results*

Based on the models of these mechanisms, we first present and analyze their numerical results (produced with MATLAB) under the above scenarios of a simplified roadside wireless sensor network.

Figure 14a shows the amount of probed contact capacity (ζ) and Fig. 14b shows the energy cost for per-unit of the probed contact capacity (ρ) during an epoch when the energy budget is low (i.e. $\Phi_{max} = \Gamma/1000$). These plots indicate that in both metrics, SNIP-RH performs much better than SNIP-AT and its performance is equivalent to SNIP-OPT. When $\zeta_{target} \leq 24$ s, SNIP-AT cannot probe the necessary contacts under the energy budget, but SNIP-RH still can energy efficiently probe the necessary contacts. When $\zeta_{target} > 24$ s, although none of these mechanisms can probe the necessary contacts, compared with SNIP-AT, SNIP-RH can probe much more contact capacity with a much lower energy cost for per-unit of probed contact capacity.

Figure 14c and d shows the results when the energy budget of a sensor node is high (i.e. $\Phi_{max} = \Gamma/100$). These plots indicate that, when Φ_{max} is large and $\zeta_{target} \leq 48$ s, SNIP-RH can probe the necessary contacts much more energy efficiently than SNIP-AT. They also indicate that when $\zeta_{target} = 56$ s, SNIP-RH cannot probe the targeted contact capacity. SNIP-AT and SNIP-OPT can achieve this target in this case, but they have a higher energy cost for per-unit of probed contact capacity. Depending on the application, it may be worthwhile to use a larger d_{rh} and/or mark more time-slots as rush hours for increasing the amount of contact capacity probed by SNIP-RH. This issue will be studied in the future.

4.4.2. *Simulation results*

To validate the above analysis, SNIP-AT, SNIP-OPT and SNIP-RH are all implemented in Contiki-OS and simulated with COOJA under all cases studied in the above numerical analysis. In the following simulations, both α and I' follow a normal distribution with small deviation (a 10th of the mean). The sensor data are generated with a constant rate derived from ζ_{target}. The duty-cycle used by SNIP-AT and the scheduling

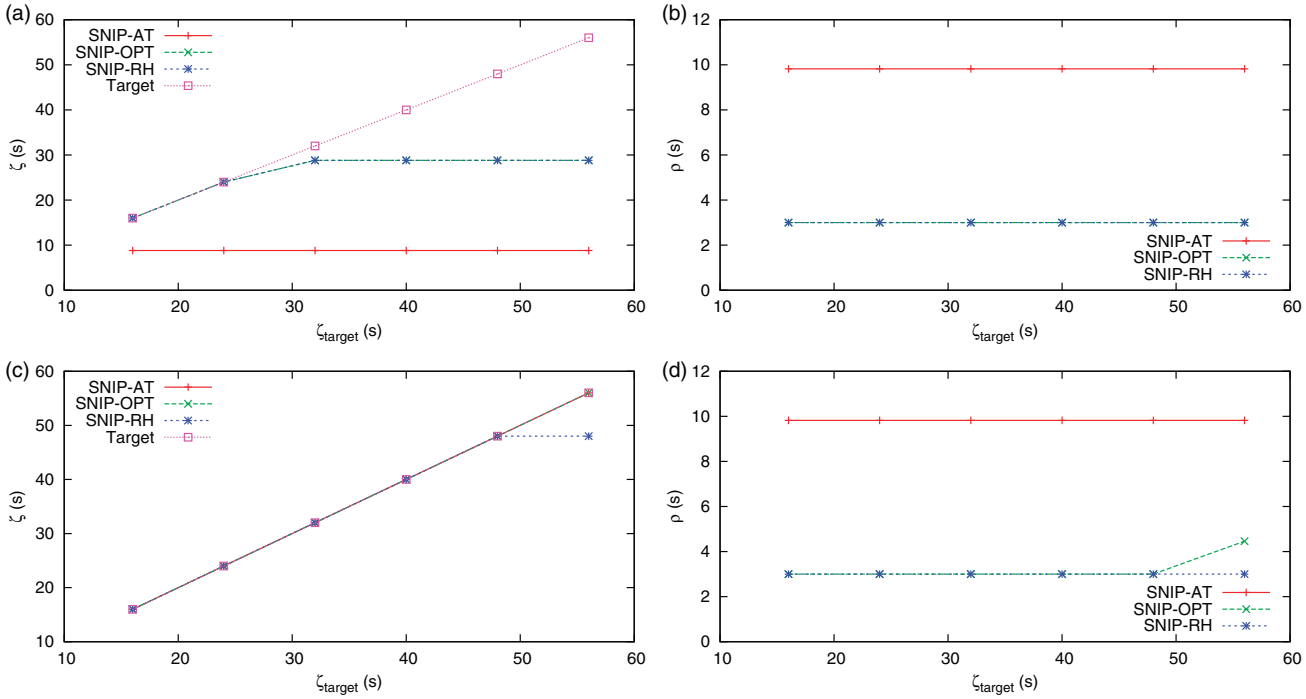

FIGURE 14. Analysis results of SNIP scheduling mechanisms. (**a**) Probed contact capacity ($\Phi_{max} = \Gamma/1000$), (**b**) per-unit cost ($\Phi_{max} = \Gamma/1000$), (**c**) probed contact capacity ($\Phi_{max} = \Gamma/100$) and (**d**) per-unit cost ($\Phi_{max} = \Gamma/100$).

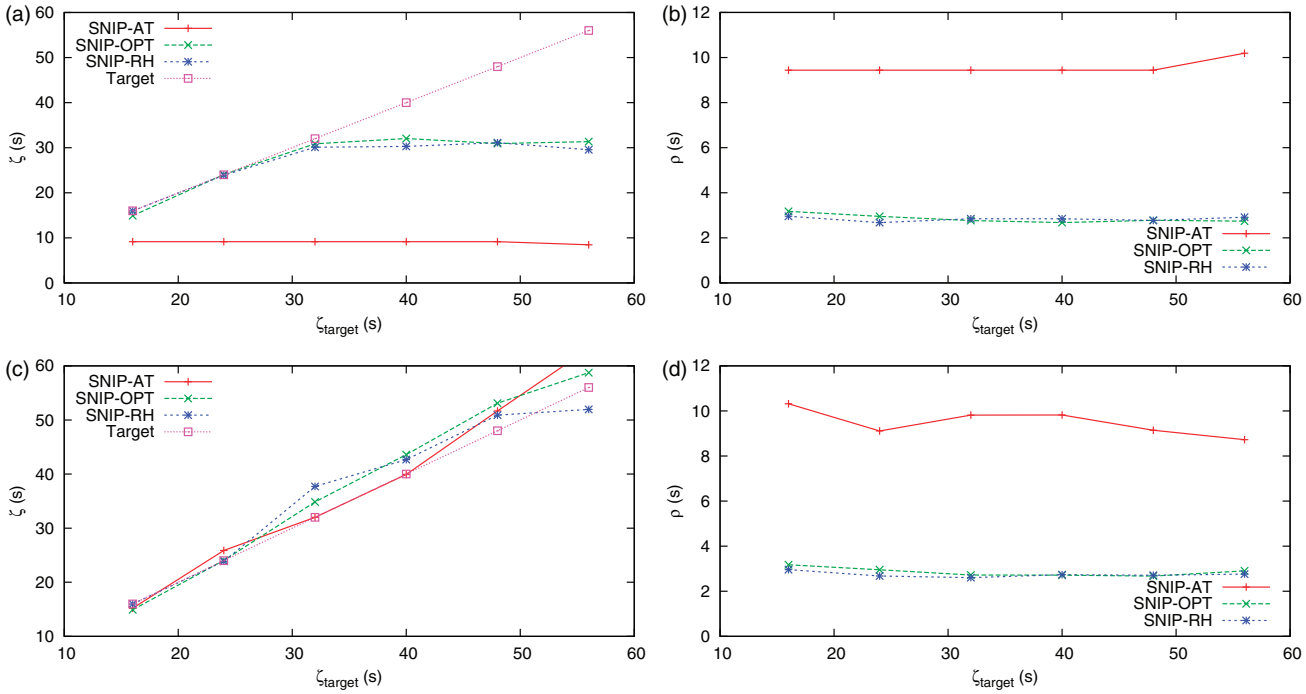

FIGURE 15. Simulation results of SNIP scheduling mechanisms. (**a**) Probed contact capacity ($\Phi_{\max} = \Gamma/1000$), (**b**) per-unit cost ($\Phi_{\max} = \Gamma/1000$), (**c**) probed contact capacity ($\Phi_{\max} = \Gamma/100$) and (**d**) per-unit cost ($\Phi_{\max} = \Gamma/100$).

plan used by SNIP-OPT are calculated with MATLAB and are incorporated into the code.

For each experiment, these mechanisms have been simulated for 2 weeks and the average results for one epoch are plotted. Figure 15 shows the results under various scenarios. These plots indicate that although there is a lot of variance in simulation results, the conclusions drawn from numerical analysis are still correct.

4.5. Evaluation results: Nokia MDC dataset

To evaluate SNIP-RH under more realistic scenarios, trace-based simulations are carried out below based on the full Nokia MDC dataset that contain GPS readings of almost 200 participants in the course of 1+ years [17, 18]. Two locations in an urban area of Lausanne are selected and we assume that a sensor node is deployed in each location. The contact arrival sequence of a sensor node is then generated based on the GPS readings within its communication range. Since a sensor node might not be deployed exactly on the route of a smartphone user, the contact length is calculated through dividing a distance that is much shorter than two times of its communication range, by the speed of a GPS reading. More specifically, 5 m is used in this paper.

For each location, based on its contact arrival sequence generated from a Nokia MDC dataset, we determine the rush hours for this location and calculate the average of the contact length, which used to select the duty-cycle used by SNIP-RH (d_{rh}). We also calculate the optimal duty-cycle for SNIP-AT iteratively. For SNIP-OPT, due to the large number of contacts with varying length, the huge solution space (one duty-cycle per hour) and the non-linear relationship between the used duty-cycle and the probed contact capacity, it is very difficult to calculate the scheduling for SNIP-OPT. Thus, we will compare SNIP-RH with just SNIP-AT in the following simulations.

Since the overall contact capacity is higher than the above simple roadside scenario, $\Phi_{\max}$ is just set to $\Gamma/1000$. And ζ_{target} varies from 16 to 64 s for studying how well these mechanisms perform under different situations. For each experiment, these mechanisms have been simulated for 20 days and the average results for one epoch are plotted.

Figure 16 shows the results of two locations under various situations. These plots indicate that the results are similar to the simple roadside scenario and SNIP-RH performs better than SNIP-AT. Compared with the roadside scenario, the performance gap between SNIP-AT and SNIP-RH is smaller. The main reason is that in the traces generated based on Nokia MDC Dataset, the contact length has larger variance. In other words, there are a few long contacts and they can be probed successfully even by SNIP-AT with a small duty-cycle. Note that compared with determining rush hours and the duty-cycle used by SNIP-RH, it is much harder for a sensor node to find the optimal duty-cycle for SNIP-AT under the dynamic and noisy environments of the real world.

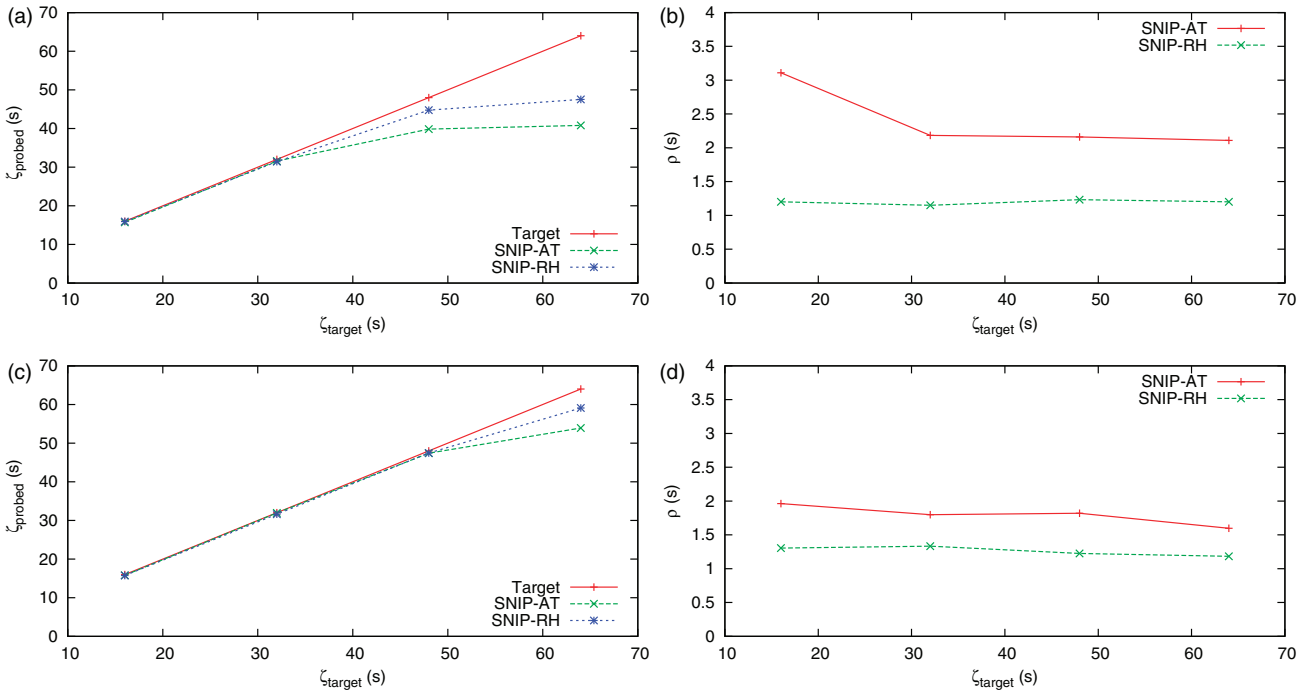

FIGURE 16. SNIP scheduling mechanisms with a Nokia MDC dataset. (**a**) Probed contact capacity (location 1), (**b**) per-unit cost (location 1), (**c**) probed contact capacity (location 2) and (**d**) per-unit cost (location 2).

4.6. Discussion

In the above evaluations, we assume that a sensor node is provided with rush hour information when deployed. When deploying in the real world, a sensor node could identify rush hours autonomously. To achieve this, a sensor node could first run SNIP-AT for a small number of epochs to learn the temporal locality of human mobility. It can then use SNIP-RH to improve the performance of contact probing. Considering that a sensor node only needs to learn the order of these time-slots' contact capacity, the learning phase could be short and the used duty-cycle could be very small. Hence, rush hours could be identified quickly and energy efficiently. Furthermore, in some environments, rush hours do have some seasonal differences [10, 27]. In this case, SNIP can be activated not only during rush hours with a high duty-cycle. It should also be activated during other times with a very low duty-cycle for tracking the seasonal shift of rush hours. In the future, we will evaluate the performance of this kind of autonomous SNIP-RH.

The contact arrival process also does not change in the above evaluations. However, the amount of contact capacity tends to vary a lot in the real world. In dynamic environments, SNIP-AT seems to be promising since SNIP is activated at all times and the environment can be learned continuously. However, to adjust its duty-cycle according to the environment and ζ_{target} (the target of probed contact capacity), SNIP-AT needs to accurately learn the contact arrival process of an epoch, both contact arrival frequency and contact length distribution. Considering that the expected applications are low data rate, it is very hard to accurately learn the time-varying contact arrival process with just a few samples (the probed contacts). The smaller ζ_{target} is, the worse is the situation. Hence, a sensor node may react based on inaccurate information and this kind of adaptive SNIP-AT may not work well in dynamic environments. As discussed in Section 4.2, SNIP-OPT also cannot work well in dynamic environments. It is even harder for a sensor node to learn the varying contact arrival process for each time-slot.

As for SNIP-RH, we argue that it could work well. Although the amount of a time-slot's contact capacity varies a lot in different epochs, the mobility pattern is invariant in the long term and rush hours will seldom change. In the case where contact capacity in rush hours is high enough to support ζ_{target}, SNIP-RH is not sensitive to the variance of the amount of contact capacity. When contact capacity becomes less, a sensor node just need to run SNIP for a longer time in rush hours and the necessary contacts still can be probed.

5. RELATED WORK

Owing to the limited computing capability and storage size of sensor nodes, sensor data are normally sent to an application server for further processing. In the classical wireless sensor network, a dedicated static base station is used to connect

sensor nodes with the application server [28]. Although sensor data can be collected continuously, the cost of the static base stations can be prohibitive under the scenarios considered in this paper. In [1–8, 29], the use of resource-rich mobile nodes was proposed to move around in the deployed area and collect data from sensor nodes within the range of one or multiple hops. In some of them [2, 6, 7, 29], the uncontrollable mobile nodes are used to collect data opportunistically. However, none of them investigated the challenges faced by contact probing in the context of opportunistic data collection with smartphones.

In addition to static sensor nodes, mobile sensor nodes have also been proposed to sense the environment, and data dissemination mechanisms among these mobile nodes have been well investigated in [30, 31]. This topic is beyond the scope of this paper.

As a basic operation for mobile computing, contact probing has been extensively studied in many other contexts. In the following subsections, we will introduce the related work in two categories, i.e. how to carry out contact probing and how to adjust the frequency of contact probing.

5.1. Contact probing methods

Contact probing has been studied for delay-tolerant applications and Bluetooth-based opportunistic applications [32, 33]. Unlike contact probing in opportunistic data collection, all nodes in these applications are similar to each other and all of them will broadcast beacons independently.

In many RFID applications [34, 35], passive RFID tags without any energy consumption are attached to objects so that they can be found and tracked by RFID readers. However, the corresponding RFID readers are too expensive, clumsy and power hungry for mobile nodes in opportunistic data collection. The reading/detection range is also too short.

In [3, 36], a mobile node with controllable mobility has been used to collect data from a sensor node. The mobile node first moves to a sensor node, collects all data from this node and moves to another sensor node. Before collecting data, a mobile node will first activate a sensor node through light, magnetic or the second low-power radio. Hence, some additional hardware components are needed for these schemes. In [1], it is assumed that sensor nodes can predict the arrival of mobile nodes. The radio of a sensor node is then turned on only when a mobile node will come soon. Hence, a sensor node can be found by a mobile node easily and energy efficiently. However, the arrival of mobile nodes cannot be predicted accurately when mobile nodes are uncontrollable and this scheme cannot be used by a sensor node to determine when to turn on the radio for contact probing. Instead, the predictability of rush hours is exploited in SNIP-RH when determining the duty-cycle used for contact probing.

Low-power MAC protocols, such as B-MAC [37] and X-MAC [38], have also been used for finding the peer to communicate with. However, in opportunistic data collection with smartphones, a mobile node does not know the position of a sensor node. Hence, it cannot know when to transmit the preamble and cannot decide the preamble's length. In addition, when collecting data from a sensor node, the throughput of these MAC protocols is too low for opportunistic data collection with uncontrollable mobility.

In [20], MNIP mechanisms have also been studied in the context of opportunistic data collection. Their shortcomings have been discussed in Section 3. Similar to SNIP, low-power listening proposed in [39, 40] also lets a sensor node periodically broadcast beacons to declare its presence. Through low-power listening, the sink node in Koala [40] can then activate the whole network hop by hop. In [36], it is also proposed to use a mobile robot with controlled mobility to collect data from sensor nodes running low-power listening. However, low-power listening has not been studied as a contact probing mechanism in the context of opportunistic data collection with uncontrollable mobility of smartphone users.

5.2. Adaptive contact probing

Adaptive contact probing has also been studied in Bluetooth-based opportunistic applications and other delay-tolerant applications [32, 33, 41]. The radio of the nodes in these applications consumes much more energy in transmitting mode. Through tuning the probing frequency, these proposals try to achieve better tradeoff between the probability of missing a contact and the energy consumed by contact probing. The characteristics of the contact arrival process are first studied and the adaptive rules are designed accordingly. For instance, the self-similarity of the contact arrival process among Bluetooth phones had been observed in [32] and the authors propose to increase the probing rate abruptly once a new contact is seen.

Instead of maintaining a low contact miss ratio, contact probing in opportunistic data collection tries to probe the necessary contacts energy efficiently, and the contact miss ratio can be large when ζ_{target} is small and contacts in the environment are abundant. Furthermore, the characteristics of the contact arrival process cannot be utilized due to the following reasons. First, sensor nodes are deployed at different places and their contacts with mobile nodes may follow different patterns [10]. Second, due to the low duty-cycle used by contact probing and the small memory of a sensor node, the node may not be able to learn the characteristics of its contact arrival process in an autonomous and timely manner.

Reinforcement learning has also been used to decide the duty-cycle used by a sensor node for contact probing [27, 42, 43]. However, a sensor node can only explore a small number of states and strategies (the duty-cycle values used by this node) due to its limited resources. With a small duty-cycle, it is also challenging to recognize the state and adopt a suitable strategy in a timely manner. Hence, a sensor node must also make decisions based on the inaccurate information learned

with a small duty-cycle and the performance may be adversely affected.

When the duty-cycle must be low and the dynamic contact arrival process cannot be accurately learned online, it should be necessary and promising to exploit the long-term invariant of the environment, such as rush hours in the repeated mobility pattern of mobile nodes. This is why and how SNIP-RH is designed in this paper.

6. CONCLUSION

In this paper, the challenges faced by contact probing in the context of opportunistic data collection are investigated systematically and SNIP, a Sensor Node-Initiated Probing mechanism, is first proposed for improving the performance of contact probing when the duty-cycle of a sensor node is fixed. The analysis, simulation and testbed evaluations demonstrate that SNIP outperforms MNIP mechanisms from the literature, especially when the duty-cycle of a sensor node is low.

Influenced by the fact that the intended applications are delay-tolerant, that mobile nodes tend to follow some repeated mobility patterns and that contacts are distributed unevenly over time, SNIP-RH is also proposed in this paper to exploit the temporal locality of human mobility. The analysis and simulation results indicate that under typical roadside wireless sensor network scenarios and the more realistic scenarios based on a Nokia MDC Dataset, SNIP-RH can significantly reduce the energy consumed for probing the necessary contacts or significantly increase the probed contact capacity under a sensor node's energy budget for contact probing.

In the future, through more trace-based simulations and additional testbed experiments, we will evaluate SNIP-RH with the capability of autonomously learning and exploiting the temporal locality of smartphone users' mobility. Furthermore, we will investigate other issues that arise when smartphones act as mobile nodes, such as incentives, user privacy and data security, that are also encountered in participatory sensing [44–46].

ACKNOWLEDGEMENTS

Portions of the research in this paper used the MDC Database made available by Idiap Research Institute, Switzerland and owned by Nokia. The authors thank the Cork Constraint Computation Centre (4C) at University College Cork for the use of their computing cluster in simulations. Preliminary results were presented in the 4th International Workshop on Sensor Networks (held with IEEE ICDCS 2011) [47] and the 1st Intl Workshop on Cyber-Physical Networking Systems (held with IEEE INFOCOM 2011) [48]. The authors thank the reviewers for their insightful comments.

FUNDING

This work is supported in part by the Irish HEA PRTLI-IV NEMBES Grant and the CTVR Grant (SFI 10/CE/I 1853) from Science Foundation Ireland.

REFERENCES

[1] Chakrabarti, A., Sabharwal, A. and Aazhang, B. (2003) Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. *Proc. 2nd Int. Conf. Information Processing in Sensor Networks*, April 22–23, pp. 129–145. Springer, Palo Alto, CA, USA.

[2] Pasztor, B., Musolesi, M. and Mascolo, C. (2007) Opportunistic Mobile Sensor Data Collection with SCAR. *Proc. IEEE Int. Conf. Mobile Adhoc and Sensor Systems*, October 8–11, pp. 1–12. IEEE, Pisa, Italy.

[3] Somasundara, A.A., Kansal, A., Jea, D.D., Estrin, D. and Srivastava, M.B. (2006) Controllably mobile infrastructure for low energy embedded networks. *IEEE Trans. Mob. Comput.*, **5**, 958–973.

[4] Liu, W., Wang, J., Xing, G. and Huang, L. (2009) Throughput Capacity of Mobility-Assisted Data Collection in Wireless Sensor Networks. *Proc. 6th IEEE Int. Conf. Mobile Adhoc and Sensor Systems*, October 12–15, pp. 70–79. IEEE, Macau, China.

[5] Zhao, M. and Yang, Y. (2009) Bounded Relay Hop Mobile Data Gathering in Wireless Sensor Networks. *Proc. 6th IEEE Int. Conf. Mobile Adhoc and Sensor Systems*, October 12–15, pp. 373–382. IEEE, Macau, China.

[6] Li, Z., Li, M., Wang, J. and Cao, Z. (2011) Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks. *Proc. IEEE INFOCOM*, April 10–15, pp. 2246–2254. IEEE, Shanghai, China.

[7] Shah, R.C., Roy, S., Jain, S. and Brunette, W. (2003) Data Mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks. *Proc. 1st IEEE Int. Workshop on Sensor Network Protocols and Applications*, May 11, pp. 30–41. IEEE, Anchorage, AK, USA.

[8] Wang, W., Srinivasan, V. and Chua, K.-C. (2005) Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. *Proc. 11th Annual Int. Conf. Mobile Computing and Networking*, August 28–September 2, pp. 270–283. ACM, Cologne, Germany.

[9] Wu, X., Sreenan, C.J. and Brown, K.N. (2011) A Shared Opportunistic Infrastructure for Long-lived Wireless Sensor Networks. *Proc. 2nd Int. ICST Conf. Mobile Lightweight Wireless Systems Workshops*, May 9–10, pp. 330–337. Springer, Bilbao, Spain.

[10] Wu, X., Brown, K.N. and Sreenan, C.J. (2013) Analysis of smartphone user mobility traces for opportunistic data collection in wireless sensor networks. *Pervasive Mob. Comput.*, **9**, 881–891.

[11] Carroll, A. and Heiser, G. (2010) An Analysis of Power Consumption in a Smartphone. *Proc. 2010 USENIX Conf. USENIX Annual Technical Conf.*, June 22–25. USENIX Association, Boston, MA, USA.

[12] Polastre, J., Szewczyk, R. and Culler, D. (2005) Telos: Enabling Ultra-Low Power Wireless Research. *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, April 25–27, pp. 364–369. IEEE Press, Los Angeles, CA, USA.

[13] Dunkels, A., Gronvall, B. and Voigt, T. (2004) Contiki—A Lightweight and Flexible Operating System for Tiny Networked Sensors. *Proc. 29th IEEE Int. Conf. Local Computer Networks*, November 16–18, pp. 455–462. IEEE, Tempa, FL, USA.

[14] Osterlind, F., Dunkels, A., Eriksson, J., Finne, N. and Voigt, T. (2006) Cross-Level Sensor Network Simulation with COOJA. *Proc. 31st IEEE Conf. Local Area Network*, November 14–16, pp. 641–648. IEEE, Tempa, FL, USA.

[15] Levis, P. and Gay, D. (2009) *TinyOS Programming*. Cambridge University Press, Cambridge.

[16] Gonzalez, M.C., Hidalgo, C.A. and Barabasi, A.-L. (2008) Understanding individual human mobility patterns. *Nature*, **453**, 779–782.

[17] Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D. and Laurila, J. (2010) Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. *Proc. 7th Int. Conf. Pervasive Services*, July 13–15. ACM, Berlin, Germany.

[18] Laurila, J.K. *et al.* (2013) From big smartphone data to worldwide research: the mobile data challenge. *Pervasive Mob. Comput.*, **9**, 752–771.

[19] Osterlind, F., Wirstrom, N., Tsiftes, N., Finne, N., Voigt, T. and Dunkels, A. (2010) StrawMAN: Making Sudden Traffic Surges Graceful in Low-Power Wireless Networks. *Proc. 6th Workshop on Hot Topics in Embedded Networked Sensors*, pp. 141–145. ACM, Killarney, Ireland.

[20] Anastasi, G., Conti, M. and Francesco, M.D. (2009) An Analytical Study of Reliable and Energy-Efficient Data Collection in Sparse Sensor Networks with Mobile Relays. *Proc. 6th European Conf. Wireless Sensor Networks*, February 11–13, pp. 199–215. Springer, Cork, Ireland.

[21] nRF24AP2 (2006) *Single-chip ANTTM ultra-low power wireless network solution (Product Specification v1.2)*. Tiller, Norway.

[22] Kim, D.H., Kim, Y., Estrin, D. and Srivastava, M.B. (2010) SensLoc: Sensing Everyday Places and Paths Using Less Energy. *Proc. 8th ACM Conf. Embedded Networked Sensor Systems*, November 3–5, pp. 43–56. ACM, Zurich, Switzerland.

[23] Zhuang, Z., Kim, K.-H. and Singh, J.P. (2010) Improving Energy Efficiency of Location Sensing on Smartphones. *Proc. 8th Int. Conf. Mobile Systems, Applications, and Services*, June 15–18, pp. 315–330. ACM, San Francisco, CA, USA.

[24] Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J. and Banchs, R. (2010) Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system. *Pervasive Mob. Comput.*, **6**, 455–466.

[25] Ross, C., Guensler, R., Washington, S. and LeBlanc, D. (1995) Temporal Distributions of Engine Starts, Hot Soaks, and Modal Operating Fractions Across Three Cities. *The Emission Inventory: Programs and Progress (A Specialty Conf. Sponsored by the US Environmental Protection Agency and the Air and Waste Management Association),* Research Triangle Park, NC, USA.

[26] Cain, A., Burris, M.W. and Pendyala, R.M. (2001) Impact of variable pricing on temporal distribution of travel demand. *Transp. Res. Rec.*, **1**, 36–43.

[27] Dyo, V. and Mascolo, C. (2007) A Node Discovery Service for Partially Mobile Sensor Networks. *Proc. 2nd Int. Workshop on Middleware for Sensor Networks*, November 26–30, pp. 13–18. ACM, Newport Beach, CA, USA.

[28] Akyildiz, I., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless sensor networks: a survey. *Comput. Netw.*, **38**, 393–422.

[29] Park, U. and Heidemann, J. (2011) Data Muling with Mobile Phones for Sensornets. *Proc. 9th ACM Conf. Embedded Networked Sensor Systems*, November 1–4, pp. 162–175. ACM, Seattle, WA, USA.

[30] Lee, U., Magistretti, E., Gerla, M., Bellavista, P. and Corradi, A. (2009) Dissemination and harvesting of urban data using vehicular sensing platforms. *IEEE Trans. Veh. Technol.*, **58**, 882–901.

[31] de Freitas, E.P., Heimfarth, T., Wagner, F.R., Pereira, C.E. and Larsson, T. (2013) Exploring geographic context awareness for data dissemination on mobile ad hoc networks. *Ad Hoc Netw.*, **11**, 1746–1764.

[32] Wang, W., Srinivasan, V. and Motani, M. (2007) Adaptive Contact Probing Mechanisms for Delay Tolerant Applications. *Proc. 13th Annual ACM Int. Conf. Mobile Computing and Networking*, September 9–14, pp. 230–241. ACM, Montreal, QC, Canada.

[33] Drula, C. Fast and energy efficient neighbour discovery for opportunistic networking with bluetooth. Master Thesis, University of Toronto, Toronto, Canada.

[34] Holzinger, A., Schwaberger, K. and Weitlaner, M. (2005) Ubiquitous Computing for Hospital Applications: RFID-Applications to Enable Research in Real-Life Environments. *Proc. 29th Annual Int. Computer Software and Applications Conf.*, July 26–28. IEEE, Edinburgh, UK.

[35] Niederman, F., Mathieu, R.G., Morley, R. and Kwon, I.-W. (2007) Examining RFID applications in supply chain management. *Commun. ACM*, **50**, 93–101.

[36] Tekdas, O., Isler, V., Lim, J.H. and Terzis, A. (2009) Using mobile robots to harvest data from sensor fields. *IEEE Wirel. Commun.*, **16**, 22–28.

[37] Polastre, J., Hill, J. and Culler, D. (2004) Versatile Low Power Media Access for Wireless Sensor Networks. *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems*, November 3–5, pp. 95–107. ACM, Baltimore, MD, USA.

[38] Buettner, M., Yee, G.V., Anderson, E. and Han, R. (2006) X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Sensor Networks. *Proc. 4th ACM Conf. Embedded Networked Sensor Systems*, October 31–November 3, pp. 307–320. ACM, Boulder, CO, USA.

[39] Ye, W., Silva, F. and Heidemann, J. (2006) Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. *Proc. 4th ACM Conf. Embedded Networked Sensor Systems*, October 31–November 3, pp. 321–334. ACM, Boulder, CO, USA.

[40] Musaloiu-E, R., Liang, C.-J.M. and Terzis, A. (2008) Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. *Proc. 7th Int. Conf. Information Processing in Sensor Networks*, April 22–24, pp. 421–432. IEEE Computer Society, St. Louis, MO, USA.

[41] Dyo, V. *et al.* (2010) Evolution and Sustainability of a Wildlife Monitoring Sensor Network. *Proc. 8th ACM Conf. Embedded Networked Sensor Systems*, November 3–5, pp. 127–140. ACM, Zurich, Switzerland.

[42] Dyo, V. and Mascolo, C. (2008) Efficient Node Discovery in Mobile Wireless Sensor Networks. *Proc. 4th IEEE Int. Conf. Distributed Computing in Sensor Systems*, June 11–14, pp. 478–485. Springer, Santorini Island, Greece.

[43] Francesco, M.D., Shah, K.U., Kumar, M.J. and Anastasi, G. (2010) An Adaptive Strategy for Energy-Efficient Data Collection in Sparse Wireless Sensor Networks. *Proc. 7th European Conf. Wireless Sensor Networks*, February 17–19, pp. 322–337. Springer Coimbra, Portugal.

[44] Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S. and Srivastava, M.B. (2006) Participatory Sensing. *World Sensor Web Workshop (in conjunction with ACM SENSYS)*, October 31. ACM, Boulder, CO, USA.

[45] Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M. and Triandopoulos, N. (2008) Anonysense: Privacy-Aware People-Centric Sensing. *Proc. 6th Int. Conf. Mobile Systems, Applications, and Services*, June 17–20, pp. 211–224. ACM, Breckenridge, CO, USA.

[46] Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T. and Campbell, A. (2010) A survey of mobile phone sensing. *IEEE Commun. Mag.*, **48**, 140–150.

[47] Wu, X., Brown, K.N. and Sreenan, C.J. (2011) SNIP: A Sensor Node-Initiated Probing Mechanism for Opportunistic Data Collection in Sparse Wireless Sensor Networks. *Proc. 1st Int. Workshop on Cyber-Physical Networking Systems*, April 10–15, pp. 726–731. IEEE, Shanghai, China.

[48] Wu, X., Brown, K.N. and Sreenan, C.J. (2011) Exploiting Rush Hours for Energy-Efficient Contact Probing in Opportunistic Data Collection. *Proc. 31st Int. Conf. Distributed Computing Systems Workshops*, June 20–24, pp. 240–247. IEEE, Minneapolis, MN, USA.