CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Data pre-forwarding for opportunistic data collection in wireless sensor networks
Authors	Wu, Xiuchao;Brown, Kenneth N.;Sreenan, Cormac J.
Publication date	2012-06
Original Citation	Wu, X., Brown, K. N. and Sreenan, C. J. (2012) 'Data pre-forwarding for opportunistic data collection in wireless sensor networks', Ninth International Conference on Networked Sensing (INSS), Antwerp, Belgium, 11-14 June, pp. 1-8. doi: 10.1109/INSS.2012.6240555
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/document/6240555 - 10.1109/INSS.2012.6240555
Rights	© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-06 20:43:50
Item downloaded from	https://hdl.handle.net/10468/8984

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Data Pre-Forwarding for Opportunistic Data Collection in Wireless Sensor Networks

Invited Paper

Xiuchao Wu, Kenneth N. Brown, and Cormac J. Sreenan
Department of Computer Science, University College Cork, Ireland
Email:{x.wu, k.brown, cjs}@cs.ucc.ie

***Abstract*—In many potential wireless sensor network applications, the cost of the base station infrastructure can be prohibitive. Instead, we consider the use of mobile devices carried by people in their daily life to collect sensor reports opportunistically. Considering that human mobility demonstrates strong spatial locality and sensor nodes need to be deeply duty-cycled for longevity, data pre-forwarding (DPF), in which sensor reports are forwarded to sensor nodes visited by people more frequently even though they are not currently being visited, should be a promising scheme to improve the performance of opportunistic data collection.**

In this paper, a distributed DPF mechanism is proposed to exploit the spatial locality of human mobility in the context of opportunistic data collection. The communication protocol is first carefully designed so that sensor nodes could rendezvous and communicate with their neighbors and mobile nodes energy efficiently. A simple heuristic algorithm is then designed so that sensor nodes could decide the number of sensor reports exchanged with their direct neighbors for improving network throughput under the energy constraint of sensor nodes. The distributed DPF mechanism has been implemented in Contiki-OS and evaluated with Cooja. Evaluation results indicate that this proposal significantly outperforms the default approach (with no data pre-forwarding).

I. 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 (static) sensor nodes sparsely deployed in a large area and their sensor reports are inherently delay tolerant, since the response (if any) requires human intervention over long time scales. For example, analysis of environmental monitoring data is rarely urgent, and meter readings for billing purposes can be delayed by weeks. Due to environmental constraints and/or cost issues, these sparse wireless sensor networks also tend to be partitioned, and deploying large numbers of static sink nodes for collecting sensor data from these sensor nodes would incur prohibitive costs in terms of deployment, maintenance, and data back-haul. Note that sensor report and sensor data are used interchangeably in this paper.

In [7][12][15][16][18][21], the use of resource-rich mobile nodes (mobile sinks or mobile relays) was 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 or part of the network, and their mobility can be either controlled or not. In this paper, we assume that mobility is not controlled and thus sensor reports are collected opportunistically. Mobile nodes could be specific devices carried by people or animals who move around the deployed area for purposes other than data collection. More interestingly, as illustrated in figure 1, they could also be the increasingly ubiquitous smart phones (installed with the corresponding radio and software) carried by people who happen to pass through the deployed area in their daily life.

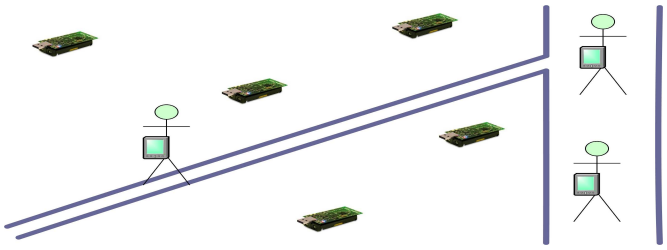


Fig. 1: Data Collection through Smart Phones

Apart from the benefits of adopting mobile sinks discussed in [15][16] (increased network reliability through removing the dependency on static sink nodes, energy efficient data collection through reducing the distance from data source to data sink, extended network lifetime through removing hot-spots near the static sink nodes, etc.), the cost of data collection can also be reduced significantly through exploiting the uncontrolled, but free human mobility. Although data delivery latency could be long in opportunistic data collection [15], there are many promising wireless sensor network applications which are delay-tolerant and it is worthwhile to improve the performance of opportunistic data collection.

However, in order to exploit human mobility for opportunistic data collection there are still many challenges to be solved, especially when sensor nodes need to be aggressively duty-cycled (e.g. duty cycle is lower than 0.1%). The combination of deep duty-cycling and uncontrolled mobility makes contact probing a challenging task, in which a mobile node and a sensor node should find each other energy efficiently and timely so that more data can be collected during one encounterance. In [19][20], it was established that the duty-cycled sensor nodes should be responsible for broadcasting beacons and the temporal locality of human mobility should be learned and exploited when determining the frequency of broadcasting beacons. In this paper, we focus on the challenges

brought by the strong spatial locality of human mobility, which has been identified in previous studies [5][6].

Due to the spatial locality of human mobility, sensor nodes deployed at different locations tend to be visited by people with different probabilities, and so they have different contact capacities - the amount of data that could be collected from a sensor node directly by mobile nodes. Furthermore, even in sparsely-deployed wireless sensor networks, there may exist many connected components and sensor reports could flow among nodes that belong to the same connected component. Thus, to improve network throughput, sensor nodes that are seldom visited or are never visited should send their sensor reports to other nodes with higher contact capacity.

The obvious way to do this is to let a mobile node collect sensor reports through multi-hop communication when it is visiting the deployment area. When a sensor node is being visited and all its sensor reports have been collected, the mobile node could then collect sensor reports from other nodes through this sensor node. However, this scheme becomes unsuitable in the context of opportunistic data collection. Due to the uncontrolled human mobility, the contact length - the period that a mobile node can communicate with a sensor node - could be as short as a few seconds. Considering that sensor nodes also need to be deeply duty-cycled for longevity, during the period that a sensor node is visited, its neighbors may not wake up and their sensor reports cannot be collected through multi-hop communication. Hence, Data Pre-Forwarding (DPF), in which sensor reports are forwarded to sensor nodes with higher contact capacity even though they are not currently being visited, should be a promising scheme for improving the performance of opportunistic data collection.

In this paper, a distributed DPF mechanism is proposed to exploit the spatial locality of human mobility in the context of opportunistic data collection. The communication protocol is first carefully designed so that sensor nodes could rendezvous and communicate with their neighbors and mobile nodes energy efficiently. A simple heuristic algorithm is then designed so that sensor nodes could decide the number of sensor reports exchanged with their direct neighbors for improving network throughput under the energy constraint of sensor nodes. The distributed DPF mechanism has been implemented in Contiki-OS [3] and evaluated with Cooja [10]. Simulation results indicate that this proposal significantly improves on the default case (with no data pre-forwarding).

The paper is organized as follows. Section II first presents the distributed DPF mechanism. Evaluation results are then presented and analyzed in section III. Finally, section IV discusses related work and section V concludes this paper.

II. The Distributed DPF Mechanism

Considering that human mobility tends to follow some repeated patterns [5], data pre-forwarding should be carried out periodically. T_{dpf} (the interval between two consecutive runs) should also be a multiple of Γ (the epoch length of the repeated human mobility pattern) for avoiding unnecessary data exchange caused by the bursty contact arrivals within an epoch. Since Γ is normally measured in hours or days, we set T_{dpf} to Γ and this value should be large enough to maintain a low overhead of data pre-forwarding. To carry out data pre-forwarding distributedly, the straightforward way is to periodically wake up all sensor nodes so that they can exchange node states and sensor reports simultaneously. However, this scheme becomes infeasible since there is no static sink node and sensor nodes must be deeply duty-cycled.

First, it is very challenging to wake up all sensor nodes energy efficiently. We may first synchronize all sensor nodes and let them wake up at the same time. However, it is hard to synchronize them efficiently since there is no static sink node and sensor nodes may not have GPS [17]. Alternatively, we may wake up sensor nodes hop by hop through the scheme used in Koala [9]. But sensor nodes are deeply duty-cycled and they must keep radio on for a long time before all nodes have woken up, especially when network diameter is large.

Second, if all sensor nodes wake up simultaneously for exchanging node states and sensor reports, there will be a lot of concurrent communications. If the existing CSMA-based protocols (X-MAC[2], Koala [9], etc.) are used, the hidden terminal problem could become severe and the existing solutions (RTS/CTS [1], etc.) are not suitable due to the large overhead. Although TDMA can be used to avoid packet collision, it is not easy or cheap to allocate time-slots among sensor nodes, especially when there is no static sink node [14]. Furthermore, clock drift could be large between two consecutive runs of data pre-forwarding since T_{dpf} is huge. The guard time between time-slots must be large and the energy of sensor nodes will be wasted.

Based on the above observations, we propose to let each sensor node randomly and independently choose a time at which it initiates data pre-forwarding for exchanging sensor reports with its direct neighbors. Since the network is sparsely deployed and T_{dpf} is huge, it is unlikely that two nodes within two-hop range will initiate data pre-forwarding at the same time[1]. Thus, a sensor node which initiates data pre-forwarding could be in control of the wireless channel. The initiator can then turn on its radio to rendezvous with its neighbors and coordinate with them to carry out data exchange efficiently.

In the following subsections, we will first present the details of the communication protocol used by mobile nodes and sensor nodes for rendezvous and data communication. The heuristic algorithm used by a sensor node to decide the number of sensor reports exchanged with its direct neighbors is then described in detail.

A. Rendezvous and Data Exchange

The schemes proposed in [19][20] are adopted here for contact probing, i.e., the rendezvous between a mobile node and a sensor node. The duty-cycled sensor node is responsible for broadcasting beacons and the temporal locality of

[1]The rare event in which the neighboring sensor nodes initiate data pre-forwarding simultaneously can be detected easily and be solved quickly. When some unexpected packets are received by a sensor node who is initiating data pre-forwarding, it can randomly choose its time again.

Msg./Notation	Usage
BEACON	broadcasted by a sensor node for contact probing and rendezvous with neighbors. It contains the current state of this node.
ASSOC_RSP	sent by mobile node after receiving a BEACON from a sensor node.
ASSOC_DONE	sent by sensor node after receiving ASSOC_RSP from mobile node. It indicates the completion of contact probing.
END	sent by sensor node to notify mobile node that all of its sensor reports have been uploaded.
RDV	sent by the initiator, the sensor node who is initiating data pre-forwarding, after receiving a BEACON from a neighbor. It contains the time that PLAN will be broadcasted.
RDV_ACK	sent by a neighbor after receiving a RDV from the initiator. It contains SNR (Signal to Noise Ratio) experienced by the RDV. Thus, the initiator can remove the neighbor if the link between them is too weak.
PLAN	broadcasted by the initiator after the data pre-forwarding plan has been produced based on local information. For each neighbor, it specifies when and how many sensor reports this neighbor should forward to the initiator. It also contains the time that all neighbors should have completed data forwarding.
T_{dpf}	the interval between two consecutive runs of data pre-forwarding It is set to Γ, the epoch length of the repeated human mobility pattern.
C_i	the contact capacity of node i in an epoch (the maximal number of sensor reports that can be collected directly from node i)
B_i	the buffer size of node i in the unit of sensor report
D_i	the current number of sensor reports in the buffer of node i
Φ_i	energy budget of node i, i.e., sensor reports that it can upload/receive/forward in an epoch
e_i	the remaining energy of node i for uploading/receiving/forwarding sensor reports in this epoch
T_{on}	the period that the radio is turned on for broadcasting BEACON and receiving possible response
T_{off}	the period that the radio is turned off for maintaining a low duty cycle
T_{beacon}	the interval between two consecutive beacons when a sensor node carries out contact probing. $T_{beacon} = T_{on} + T_{off}$.
T_{idle}	the threshold used by mobile node and sensor node to detect whether mobile node has moved away. It is currently set to 50ms.
T_{rdv}	the period that the radio is turned on by the initiator for rendezvousing with all of its neighbors. T_{rdv} should be longer than the largest possible value of T_{beacon}.
T_{plan}	the interval between the beginning of data pre-forwarding and the time to broadcast PLAN. $T_{plan} > T_{rdv}$ and the difference must be large enough to produce data pre-forwarding plan (the number of sensor reports exchanged with neighbors).
Υ	the threshold used to reduce the overhead of data forwarding

TABLE I: Messages and Notations Used for Contact Probing and Data Pre-Forwarding

human mobility is learned and exploited when determining the frequency of beacon broadcasting. In this paper, the beacons broadcasted for contact probing are exploited further to rendezvous and exchange states among sensor nodes. In the following subsections, we will present the details for both mobile node and sensor node using their corresponding state transition diagrams.

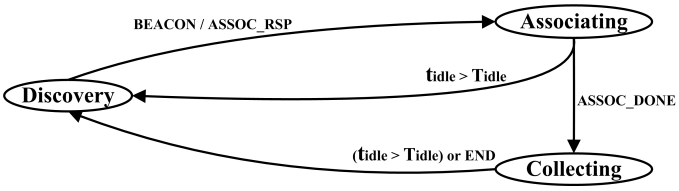


Fig. 2: State Transition Diagram of Mobile Node

1) Mobile Node: Figure 2 shows the state transition diagram of a mobile node, which moves around in an uncontrolled manner and its radio is always turned on so that it can be discovered. This is feasible since the mobile node could be equipped with abundant and rechargeable power supply [20].

When a mobile node is booted up, it first turns on its radio and enters into Discovery state. 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 sensor node, the association is completed. 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 small constant (T_{idle}), the mobile node returns back to Discovery state and is ready to be discovered again.

Table I lists the messages and notations used by mobile node and sensor node in this distributed DPF mechanism.

2) Sensor Node: Figure 3 shows the state transition diagram of a sensor node. After a sensor node is booted up, its radio is off and it is in Sleeping state. The DPF timer used to carry out data pre-forwarding periodically is started with a period that is randomly selected in the range of [0, T_{dpf}]. The beacon timer is also started to broadcast a BEACON after a short period. To support data pre-forwarding, the BEACON also contains the current states of this sensor node needed by the following heuristic algorithm.

a) Contact Probing: When the beacon timer expires, it will turn on its radio, send out a BEACON, and enter into Discovery state. If this node does not receive any response within a short period (T_{on}), it will turn off its radio, return to the Sleeping state, and start its beacon timer with a period (T_{off}) that is much longer than T_{on} for maintaining a low duty cycle. Note that T_{on} and T_{off} determine the interval between two consecutive beacons, $T_{beacon} = T_{on} + T_{off}$. Based on the current sensor node platforms, T_{on} is set to 20ms, which is enough for sending a BEACON and receiving a response from mobile node or its neighbor. T_{off} may be adjusted to learn and exploit the temporal locality of human mobility [19].

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

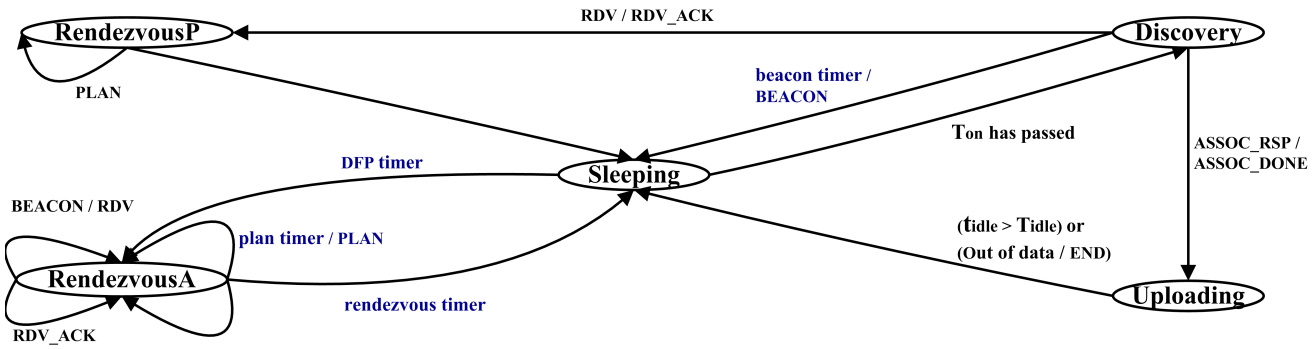


Fig. 3: State Transition Diagram of Sensor Node

timer is used for reliable data transmission, and multiple sensor reports are concatenated into one packet for reducing the overhead of packet headers. If all data had 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 T_{idle}, the sensor node will turn off its radio, return back to Sleeping state, and start its beacon timer for the next probing.

b) Data Pre-Forwarding - the initiator: When a sensor node's DPF timer expires, it first turns on its radio and enters into RendezvousA state. The DPF timer is then restarted with a period (T_{dpf}) for scheduling the next run of data pre-forwarding. This node will be referred as the initiator below.

The initiator will also start two timers, the rendezvous timer whose period is T_{rdv} and the plan timer whose period is T_{plan}. T_{rdv} is the period that the initiator keeps its radio on for rendezvousing with its neighbors. T_{rdv} should be longer than the largest possible value of T_{beacon} so that the initiator could receive a BEACON from each of its neighbors. T_{plan} is the interval between the beginning of data pre-forwarding and the time to broadcast PLAN, which instructs how the neighbors should forward data to the initiator. For each neighbor, PLAN specifies when and how many sensor reports this neighbor should forward to the initiator. It also contains the time that all neighbors should have completed data forwarding. Obviously, T_{plan} must be larger than T_{rdv} so that the data pre-forwarding plan could be produced before broadcasting PLAN. Since the following heuristic algorithm makes decisions based on the states of a few direct neighbors, data pre-forwarding plan can be produced quickly and the difference between T_{plan} and T_{rdv} can be set to a small value.

In RendezvousA state, if a BEACON is received from a neighbor, the initiator will send back a RDV which contains the time that the initiator will broadcast PLAN. If a RDV_ACK is received from this neighbor, the initiator will decide whether this neighbor should be considered based on link quality experienced by RDV and RDV_ACK. Thus, we can avoid using the asymmetric or weak links to exchange data among sensor nodes. When the rendezvous timer expires, the initiator should have collected node states from all of its direct neighbors. It will run the following heuristic algorithm to produce the data pre-forwarding plan.

When the plan timer expires, the initiator will broadcast PLAN to its neighbors for three times so that it could be more robust to packet corruption. The initiator will then turn off its radio. After that, based on PLAN, it will schedule its radio in a TDMA-like manner to receive sensor reports from its neighbors. As for data communication between the initiator and a neighbor, we adopt the same protocols used for data uploading between sensor node and mobile node. When the initiator has completed the communications with all neighbors, it will enter into Sleeping state, turn off its radio, and start the beacon timer for contact probing.

c) Data Pre-Forwarding - the neighbor: As illustrated in figure 3, after broadcasting a BEACON, a sensor node enters into Discovery state. If a RDV is received in this state, it will first send back a RDV_ACK, which contains the SNR (signal to noise ratio) experienced by the RDV. This node will then turn off its radio and enter into the RendezvousP state. This sensor node will be referred as the neighbor below. Note that a sensor node acts as both an initiator and the neighbor of other nodes (at different times).

Based on the just received RDV, the neighbor can turn on its radio at the expected time to receive PLAN from the initiator. If PLAN does not arrive as expected, the neighbor will turn off the radio, enter into Sleeping state and start the beacon timer for contact probing. If PLAN is received and this neighbor need forward sensor reports, it will schedule its radio and forward data to the initiator accordingly. After the required number of sensor reports have been forwarded, this neighbor will send END and turn off its radio. When all neighbors should have completed data forwarding (this time is included in PLAN), this neighbor will enter into Sleeping state and start the beacon timer for contact probing.

B. Heuristic Algorithm

To determine the number of sensor reports that each neighbor should forward, a heuristic algorithm is designed here. The main philosophy is that if a sensor node is visited seldom, its available buffer space will reduce since the excess sensor reports will accumulate in the buffer with the elapse of the time. Through letting each sensor node pull sensor reports from its neighbors whose available buffer space is smaller, sensor reports will flow to sensor nodes that are visited more frequently.

Considering that sensor nodes are energy constrained, it is assumed that each node i has an energy budget (Φ_i), i.e., the number of sensor reports that it can upload (to mobile nodes), receive (from neighbors), or forward (to neighbors) in an epoch. When this algorithm is carried out, the initiator has collected node states from all of its K neighbors through BEACON frames. For each neighbor i, the initiator gets to know B_i (buffer size of this neighbor), D_i (the current number of sensor reports in the buffer), and e_i (the remaining energy resource, i.e., the number of sensor reports that this neighbor still can receive, forward, or upload in the current epoch).

The initiator first calculates its current available buffer space, $AB_{Inr} = B_{Inr} - D_{Inr}$. For each neighbor i, AB_i is also calculated in the same way as AB_{Inr}. The average of the available buffer space across the initiator and the neighbors whose available buffer space is smaller than AB_{Inr} is then calculated as follows.

$$\overline{AB} = \frac{AB_{Inr} + \sum_{AB_i < AB_{Inr}} AB_i}{1 + \sum_{AB_i < AB_{Inr}} 1} \quad (1)$$

We then calculate the total number of sensor reports that these neighbors hope to forward.

$NF_{tot} = \sum_{AB_i < \overline{AB}} NF_i$. Here, $NF_i = \min((\overline{AB} - AB_i), e_i)$

The total number of sensor reports, that the initiator can receive, is $NR_{tot} = \min(AB_{Inr} - \overline{AB}, NF_{tot}, e_{Inr})$. As for the number of sensor reports that the initiator will receive from neighbor i, NR_i is calculated as follows.

$$NR_i = \begin{cases} 0 & AB_i \geq \overline{AB} \text{ or } \frac{NR_{tot} * NF_i}{NF_{tot}} < \Upsilon \\ \frac{NR_{tot} * NF_i}{NF_{tot}} & otherwise. \end{cases} \quad (2)$$

More specifically, when it is impossible to receive all sensor reports, the initiator will receive from its neighbors proportionally. When NR_i is less than a threshold (Υ), neighbor i will not forward sensor reports with aim of reducing the overhead of data transmission. In the following evaluations based on TELOSB platform [13], Υ is set to 4, i.e., the maximal number of sensor reports that can be put into a packet when Zigbee radio is used and the size of a sensor report is 30 bytes.

After NR_i is determined for all neighbors, the initiator can compose PLAN based on the bandwidth of the wireless channel. For each neighbor i, it will allocate enough time for forwarding NR_i sensor reports to the initiator. Hence, for each neighbor i that needs to forward sensor reports, the PLAN has a corresponding three-tuple: ⟨the node id of neighbor i, NR_i, the time to start to forward sensor reports⟩. It also contains the time that all neighbors should have completed data forwarding.

III. Evaluation Results

To evaluate the above distributed DPF mechanism, we have implemented it in Contiki-OS [3] and evaluated this proposal through simulations with the Cooja simulator [10]. For comparison, the naive algorithm (WO-DPF), in which sensor reports do not flow among sensor nodes, is implemented in Contiki-OS and simulated with Cooja too. Since sensor nodes prefer high throughput during data communication and low duty cycle is preferred at other times, NULLMAC of Contiki-OS is used and the radio is directly controlled by the DPF mechanism. In the simulations with Cooja, UDGM (Unit Disk Graph Medium) is used as the radio medium and the transmission range is set to 50m.

Considering that human mobility normally demonstrates a strong diurnal pattern, the epoch length (Γ) should be 24 hours and the interval between two consecutive runs of data pre-forwarding (T_{dpf}) should be a multiple of 24 hours. To get meaningful results, each simulation must also be run for a large number of epochs. However, simulations with Cooja are very slow since a machine code instruction level emulator of TELOSB mote [13] is used for each sensor node. Hence, in the following simulations, Γ and T_{dpf} will be set to 2 hours, i.e., 7200 seconds. Other parameters are also adjusted to reflect this short epoch, and the data rate and duty cycle of a simulated sensor node could become higher than the expected opportunistic data collection applications in real world.

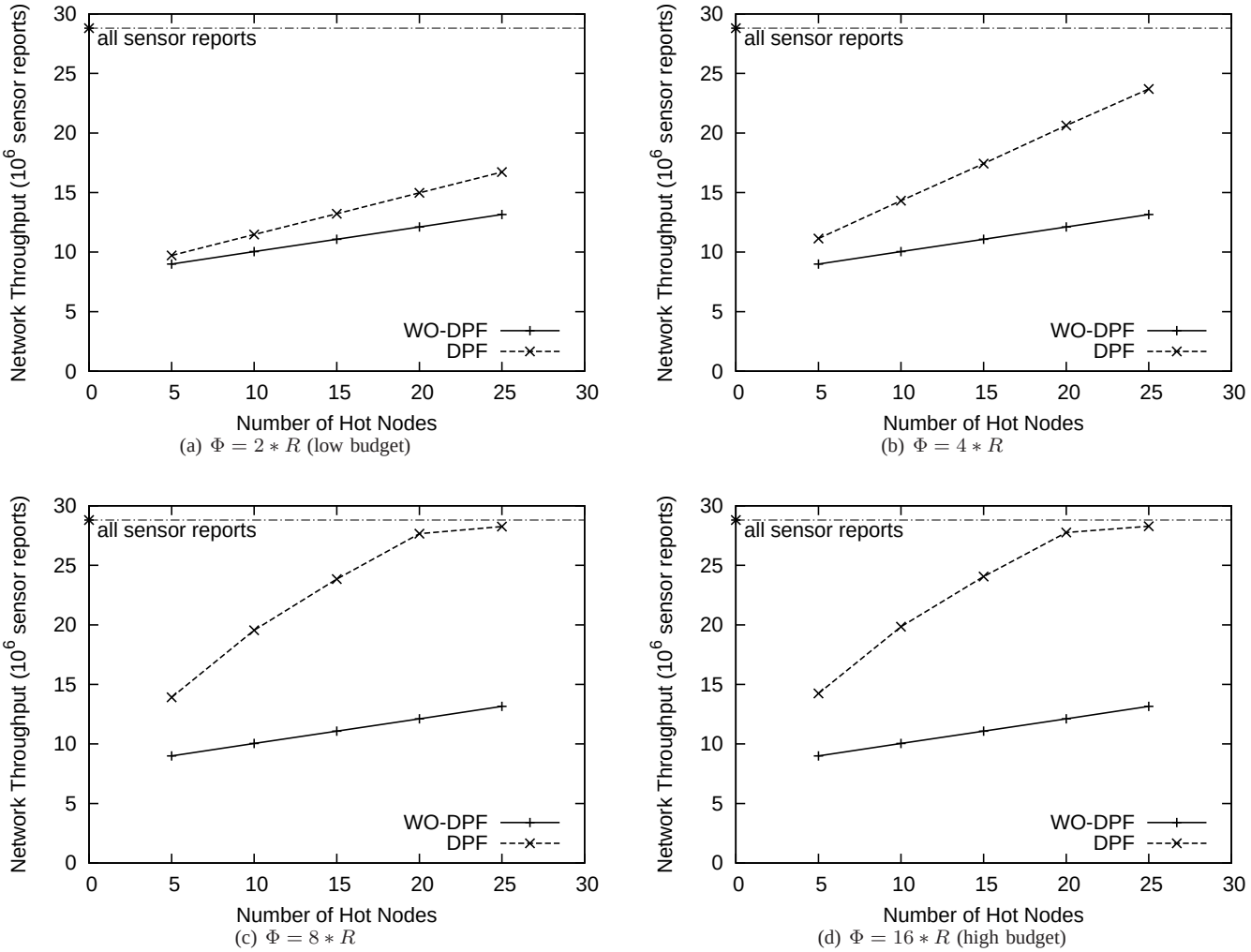
To control the contact capacity of a sensor node, rather than simulate the movement of mobile nodes directly, we implement a process on each sensor node which simulates the contact arrivals and removes the corresponding number of sensor reports from the buffer. For simplicity, we assume that there are two kinds of sensor nodes, hot nodes and cold nodes. A hot node will communicate with mobile nodes 20 times per epoch, while a cold node only has the opportunity of communicating with mobile nodes once per epoch. We assume that the size of a sensor report is 30 bytes and 400 sensor reports can be collected during each contact. This is close to the amount of data that the smart phone of a car driver can collect when it passes by a sensor node with a moderate speed and Zigbee radio is used between the smart phone and the sensor node. In the following simulations, contacts are distributed evenly in an epoch.

As for the data rate, each sensor node will generate one sensor report per 5 seconds. Thus a sensor node will generate $R = 7200/5 = 1440$ sensor reports in an epoch. T_{beacon} is also set to 5 seconds. Accordingly, T_{rdv} and T_{plan} are set to 6s and 8s, respectively. In opportunistic data collection, flash disk with large capacity should be installed on a sensor node for storing data since the uncontrolled mobile nodes may not pass by for a long time. Hence, the buffer size of a sensor node (B) is now set to 216K bytes, i.e., 7200 sensor reports can be buffered at a sensor node. Table II summarizes the parameter values used in these simulations.

Parameter	T_{dpf}	T_{on}	T_{beacon}	T_{rdv}	T_{plan}
Value	7200s	20ms	5s	6s	8s
Parameter	B (rpts)	Υ	R (rpts/epoch)	C_{hot}	C_{cold}
Value	7200	4	1440	8000	400

TABLE II: Parameter Values

In the following subsections, we will present the experimental settings and simulation results. More specifically, we will first evaluate the distributed DPF mechanism with a grid topology under various scenarios. The effects of buffer size



(a) $\Phi = 2 * R$ (low budget)
(b) $\Phi = 4 * R$
(c) $\Phi = 8 * R$
(d) $\Phi = 16 * R$ (high budget)

Fig. 4: Simulation Results with Grid Topology

and node dynamics are then studied. For each experiment, the simulation will run for a long period (400 hours, i.e., 200 epochs) to get meaningful results.

A. Grid Topology

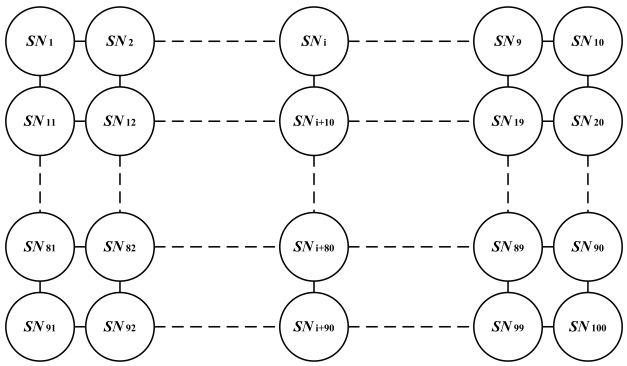


Fig. 5: A 10×10 Grid of 100 Sensor Nodes

In this group of experiments, we use a 10×10 grid of 100 sensor nodes shown in figure 5. The number of hot nodes (N_{hot}) is set to 5, 10, 15, 20, or 25, and these hot nodes spread randomly and evenly in the grid. More specifically, the grid is divided into N_{hot} continuous blocks of the same size and one node is randomly picked from each block as the hot node. Through varying N_{hot}, we can evaluate the distributed DPF mechanism under different levels of spatial locality. The energy budget of a sensor node is set according to R. More specifically, Φ will be set to $2 * R$, $4 * R$, $8 * R$, or $16 * R$ for evaluating the distributed DPF mechanism under different energy budgets of a sensor node.

Figure 4 shows the simulation results with the grid topology. The X-axis is the number of hot nodes and the Y-axis is network throughput (the total number of sensor reports collected from all sensor nodes). These plots indicate that network throughput will increase with the number of hot nodes. This is reasonable since the overall contact capacity increases with the number of hot nodes and cold nodes are much closer to hot nodes when there are more hot nodes. These plots also indicate that the distributed DPF mechanism performs quite

well. Under all cases, it collects many more sensor reports than WO-DPF, in which data pre-forwarding is not carried out at all. As illustrated in these plots, the improvements also increase with the energy budget of sensor nodes. In summary, these results indicate that data pre-forwarding is an effective mechanism to improve the performance of opportunistic data collection even when a very simple algorithm is used to make decisions based on local information.

B. Effects of Buffer Size

The same 10×10 grid topology is used in this group of experiments. To study the effects of buffer size, the energy budget is fixed to $8 * R$, N_{hot} is fixed to 15, and the buffer size is set to 3600, 5400, 7200, 9000, and 10800 sensor reports in different experiments.

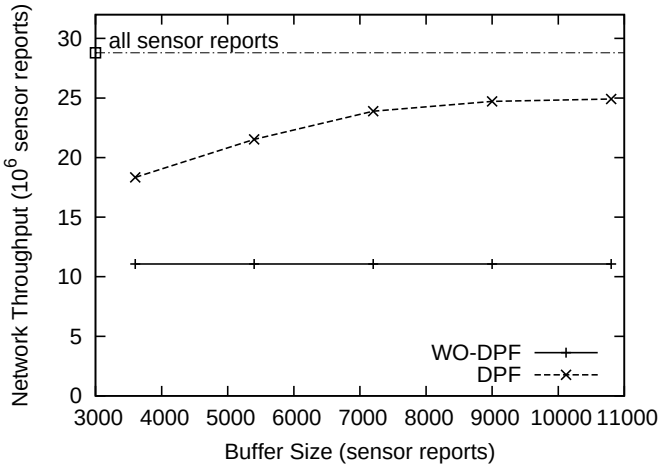


Fig. 6: Effects of Buffer Size

Figure 6 illustrates that the throughput of the DPF mechanism does increase with the buffer size. The reason is that when the buffer is large, a hot node can receive more sensor reports from its neighbors in an epoch. Considering that a large flash disk is normally used as the buffer in opportunistic data collection, the good performance of the distributed DPF mechanism can be expected.

C. Effects of Node Dynamics

In this group of experiments, the same grid topology is used, the energy budget is fixed to $8 * R$, N_{hot} is fixed to 15, and the buffer size is fixed to 7200. To study the performance of the distributed DPF mechanism under different levels of node dynamics, we will emulate node dynamics through dropping BEACONs purposely and the loss rate experienced by BEACONs is set to 0, 10%, 20%, 30%, and 40% in different experiments.

As shown in figure 7, although the throughput of the distributed DPF mechanism does decrease with the increase of node dynamics, this mechanism still performs much better than WO-DPF even when the loss rate experienced by BEACONs is as high as 40%.

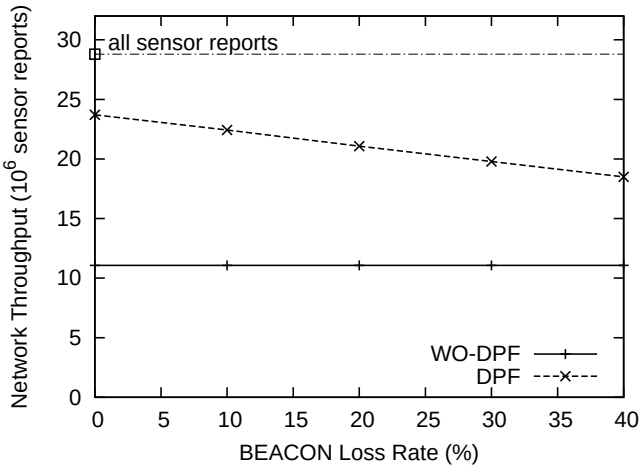


Fig. 7: Effects of Node Dynamics

IV. Related Work

In [11], the authors have proposed to use mobile phones for collecting data from static sensor nodes opportunistically or purposely. However, it has not studied the routing issues among sensor nodes for improving the throughput of data collection. In [7], the authors studied how to maintain the collection tree when the sink is moving. As stated in section I, the straightforward multi-hop data collection is not suitable since the mobility is uncontrolled and sensor nodes must be deeply duty-cycled in opportunistic data collection. We instead use data pre-forwarding to improve the network throughput.

In [4], spatial locality of the scientists' mobility has been leveraged. The domain experts assign priorities to sensor data and sensor nodes are classified based on the frequency of being visited. The hot sensor nodes visited more frequently will then build up multiple collection trees, and other nodes will forward their important sensor data to the nearest hot sensor node to reduce the latency of data collection. However, a scientist will collect all data from a sensor node once it is visited. Hence, the mobility is not totally uncontrolled and the problem studied in [4] is different with our paper.

In [19][20], contact probing has been studied by us in the context of opportunistic data collection. We found that the duty-cycled sensor nodes should be responsible for broadcasting beacons and the temporal locality of human mobility should be learned and exploited when determining the frequency of broadcasting beacons. These works are tightly related with this paper since the contact probing scheme determines the contact capacity of sensor nodes, which will affect how sensor reports flow among these nodes. Rendezvous with neighbors has also been well studied and many MAC protocols have been proposed for wireless sensor networks, such as X-MAC [2], Koala [9], and TRAMA [14]. Our proposal is similar to Koala in the way of finding the neighbors, i.e., a sensor node turns on its radio to listen for beacons from its potential neighbors.

As for data routing, our heuristic algorithm is similar to the approach proposed in [8]. We establish contact capacity

potentials that allow sensor reports to reach the sensor nodes with higher contact capacity through local greedy decisions, following information gradients. Instead of explicitly diffusing contact capacity across the network, the information is broadcasted and composed through carrying out data pre-forwarding hop-by-hop and the gradients are reflected by the values of the available buffer space.

V. Conclusion

In this paper, we studied how to use data pre-forwarding to improve the performance of opportunistic data collection. We have designed a distributed DPF mechanism in which sensor nodes could rendezvous and communicate with their neighbors and mobile nodes energy efficiently, and with a simple heuristic algorithm sensor nodes could decide the number of sensor reports exchanged with their direct neighbors for improving network throughput under the energy constraint of sensor nodes. The distributed DPF mechanism has been implemented in Contiki-OS and evaluated with Cooja simulations. Evaluation results indicate that it significantly outperforms the default approach (with no data pre-forwarding) through exploiting the spatial locality of human mobility.

In the future, we will evaluate the distributed DPF mechanism more extensively. The heuristic algorithm used by a sensor node will also be refined to improve the performance further. We also plan to model the data pre-forwarding problem and deduce the optimal result for evaluating the heuristic algorithm. Another potential area is to handle contact probing and data pre-forwarding together with the consideration of joint optimization.

Acknowledgements

This work is supported in part by the HEA PRTLI-IV NEMBES Grant. The authors thank the Cork Constraint Computation Centre (4C) at University College Cork for use of their computing cluster.

References

[1] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang. Macaw: A media access protocol for wireless lans. In *SIGCOMM*, 1997.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In *SENSYS*, 2006.

[3] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *LCN*, 2004.

[4] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pasztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef. Evolution and sustainability of a wildlife monitoring sensor network. In *Sensys*, 2010.

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

[6] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466, August 2010.

[7] Z. Li, M. Li, J. Wang, and Z. Cao. Ubiquitous data collection for mobile users in wireless sensor networks. In *IEEE INFOCOM*, 2011.

[8] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. J. Guibas. Composable information gradients in wireless sensor networks. In *IPSN*, 2008.

[9] R. Musaloiu-E, C.-J. M. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *IPSN*, 2008.

[10] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level simulation in cooja. In *EWSN*, 2007.

[11] U. Park and J. Heidemann. Data muling with mobile phones for sensornets. In *Sensys*, 2011.

[12] B. Pasztor, M. Musolesi, and C. Mascolo. Opportunistic mobile sensor data collection with scar. In *MASS*, 2007.

[13] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN*, 2005.

[14] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Sensys*, 2003.

[15] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, 2003.

[16] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8):958–973, August 2006.

[17] P. Sommer and R. Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *IPSN*, 2009.

[18] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Mobicom*, 2005.

[19] X. Wu, K. N. Brown, and C. J. Sreenan. Exploiting rush hours for energy-efficient contact probing in opportunistic data collection. In *IEEE ICDCS Workshops*, 2011.

[20] X. Wu, K. N. Brown, and C. J. Sreenan. SNIP: A sensor node-initiated probing mechanism for opportunistic data collection in sparse wireless sensor networks. In *IEEE INFOCOM Workshops*, 2011.

[21] M. Zhao and Y. Yang. Bounded relay hop mobile data gathering in wireless sensor networks. In *MASS*, 2009.