

# Practical Opportunistic Data Collection in Wireless Sensor Networks with Mobile Sinks

Shusen Yang, *Member, IEEE*, Usman Adeel, Yad Tahir, and Julie A. McCann

**Abstract**—Wireless Sensor Networks with Mobile Sinks (WSN-MSs) are considered a viable alternative to the heavy cost of deployment of traditional wireless sensing infrastructures at scale. However, current state-of-the-art approaches perform poorly in practice due to their requirement of mobility prediction and specific assumptions on network topology. In this paper, we focus on low-delay and high-throughput opportunistic data collection in WSN-MSs with general network topologies and arbitrary numbers of mobile sinks. We first propose a novel routing metric, Contact-Aware ETX (CA-ETX), to estimate the packet transmission delay caused by both packet retransmissions and intermittent connectivity. By implementing CA-ETX in the defacto TinyOS routing standard CTP and the IETF IPv6 routing protocol RPL, we demonstrate that CA-ETX can work seamlessly with ETX. This means that current ETX-based routing protocols for static WSNs can be easily extended to WSN-MSs with minimal modification by using CA-ETX. Further, by combining CA-ETX with the dynamic backpressure routing, we present a throughput-optimal scheme Opportunistic Backpressure Collection (OBC). Both CA-ETX and OBC are lightweight, easy to implement, and require no mobility prediction. Through test-bed experiments and extensive simulations, we show that the proposed schemes significantly outperform current approaches in terms of packet transmission delay, communication overhead, storage overheads, reliability, and scalability.

**Index Terms**—Opportunistic networks, sensor data collection, delay tolerant sensing, queueing networks

## 1 INTRODUCTION

WIRELESS sensing technologies have been playing an increasing important role in environment monitoring applications [1], [2], [3]. Traditional multi-hop Wireless Sensor Networks (WSNs) aims to support real-time data delivery (in hundreds of milliseconds) [4], [5], [6], [7], which may fail in large-scale sensing applications. This is because that thousands of sensor nodes would be required to ensure both the sensing coverage and multi-hop network connectivity, resulting in expensive deployment cost and network congestions around the sink. Alternatively, equipping sensing devices with cellular communication radios can also support real-time data collection, but this will lead to expensive phone bill costs (e.g., \$4,500 annual costs for 12 sensors reported in [8]) and short network lifetime caused by the required high transmission power (e.g., typically 500 mW) for long-range wireless communications.

In fact, many environment monitoring applications do not have a strict real-time requirement (e.g., in minutes or hours) such as temperature and urban noise monitoring [1], [9], [10]. For such applications, wireless devices carried by vehicles or individuals (e.g., smart phones) can act as mobile sinks to collect data from static sensors in an opportunistic way, by using

short-range wireless radios such as Bluetooth Low Energy (BLE), WiFi direct, and Zigbee [11], [12], [13]. With the increasing short-range communication capabilities of mobile devices and their huge population, WSNs with mobile sinks (WSN-MSs) [9], [10], [14], [15] have a great potential to become flexible and cost-effective solution to such delay-tolerant sensing applications.

In a WSN-MS, a sensor node can either send its data directly to a sink as it passes by, or it can send data via multi-hop routes to other sensor nodes who currently have contact with, or who will be likely in contact with a sink in the future. Therefore, how to choose the best routes to efficiently forward sensor data is a key issue for data collection in WSN-MSs. Although this topic has received a reasonable amount of research to date, most of them have limitations that affect their adoption in practice. Some approaches [9], [10], [15] require the prediction of trajectories of the sinks, which incurs considerable overheads and suffers from prediction errors or may not even be possible in large-scale practical scenarios. While other schemes, such as [14], focus on adaptively and smoothly updating routing tree structures as a sink moves in the sensing area. These schemes would suffer from heavy communications overheads in the following cases: (1) WSN-MSs with large numbers of fast moving sinks; (2) intermittently-connected WSN-MSs where a sensor node cannot reach the sink through any end-to-end path for reasonable periods of time (e.g., off-peak time in urban roads in [2]). Furthermore, due to the opportunistic availability of mobile sinks and heavy data traffic potentially produced by ubiquitous sensors, throughput performance is an important issue for data collection schemes in WSN-MSs. However, this has received very little attention in current WSN-MS research.

- S. Yang is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: cuiangmu@gmail.com.
- U. Adeel is with Intel Labs Europe, Santa Clara, CA 95054-1549. E-mail: usman.adeel@intel.com.
- Y. Tahir and J. A. McCann are with the Department of Computing, Imperial College London, London SW7 2AZ, United Kingdom. E-mail: {y.tahir11, j.mccann}@imperial.ac.uk.

Manuscript received 21 Apr. 2015; revised 7 July 2016; accepted 18 July 2016. Date of publication 28 July 2016; date of current version 31 Mar. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2595574

To overcome the limitations of existing approaches, this paper aims to develop high-throughput and low-delay opportunistic data collection approaches for *practical WSNs-MSs with general network topologies and sink mobility patterns* (e.g., connected or intermittently connected networks, arbitrary numbers of sinks, and arbitrary reasonable sink movement speeds). The contributions of this paper are summarized as follows:

1. Based on queuing analysis theory, we propose a novel routing metric Contact-Aware ETX (CA-ETX) for WSN-MSs, to effectively estimate the opportunistic link quality between sensors and mobile sinks. Specifically, CA-ETX performs an on-line measurement of the packet transmission delay over each opportunistic link, by considering both intermittent contact opportunities and the wireless unreliability during each contact (i.e., data packet retransmissions). Beside its efficiency, a major advantage of CA-ETX is that it can simultaneously work with ETX, the most popular link quality metric used by current WSN routing protocols including the defacto TinyOS routing standard CTP [5] and the IETF IPv6 routing protocol RPL [16]. By implementing CA-ETX in two popular WSN operating systems TinyOS [17] and Contiki [18], we demonstrate that CTP [5] and RPL [16] can be easily extended to support delay-tolerant data collection in WSN-MSs by using CA-ETX.
2. By integrating CA-ETX into stochastic Lyapunov optimization theory, we propose a throughput-optimal approach, Opportunistic Backpressure Collection (OBC), a joint multi-path routing and scheduling protocol for WSN-MSs. In contrast to current data collection schemes in WSN-MSs, OBC is lightweight, easy to implement, requires no mobility prediction, and can support a large number of fast moving sinks. To our knowledge, OBC is the first scheme that combines the backpressure algorithm [19] and mobility awareness for WSN-MSs. We implement OBC in both TinyOS [17] and a realistic WSN simulator Castalia [20].
3. We construct real-world experiments and extensive simulations to evaluate the performance of CA-ETX and OBC based on the IEEE 802.15.4 radio.<sup>1</sup> The evaluation results show that both delay and packet loss performance of both CTP and RPL can be significantly improved by simply adopting CA-ETX over opportunistic sensor-sink links. In addition, evaluation results demonstrate that OBC can achieve significant performance improvements in terms of end-to-end delay, storage overheads, energy efficiency, packet loss, and scalability, compared with state-of-the-art approaches.

The remainder of this paper is organized as follows: the next section discusses related work; we present our system model in Section 3; Section 4 proposes the CA-ETX metric;

Section 5 presents the OBC protocol. Implementation details and evaluation results are presented and discussed in Section 6; and finally we conclude the paper in Section 7.

## 2 RELATED WORK

*Data Collection in WSN-MSs.* There are several data collection protocols for WSN-MSs [9], [10], [14], [15], [21]. Lee et al. [15] propose a routing protocol based on information potentials [22] and a constructed mobility graph. However, this scheme requires mobility prediction and may suffer from heavy communication overheads when there are a large number of mobile sinks. WARP [23] and the routing protocol developed by Li et al. [14] are based on fast and efficient routing structure repairs, but are still limited to single mobile sink settings. Data Stashing [10] supports multiple mobile sinks but requires mobility prediction, network-wide flooding, and linear programming solving on each node, leading to large communication and computational overheads. Furthermore, none of these approaches focus on throughput performance.

*Wireless Routing Metrics.* In static wireless networks such as typical multi-hop WSNs, routing metrics like the well-known Expected Transmission Count (ETX) [24] estimate the packet transmission delays caused by link unreliability (i.e., retransmissions). In mobile networks such as Delay Tolerant Networks (DTNs), routing metrics such as inter-contact time [25], contact probabilities [26], or social network based metrics such as centrality [27] are widely used. However, all these metrics ignore the channel quality of the temporary connected wireless link. In contrast to existing routing metrics, CA-ETX is specifically designed for the WSN-MSs and can efficiently estimate packet waiting times in data buffers, which is the major delay of per-hop packet transmission in WSNs-MSs.

*Backpressure Algorithms.* [6], [19] are well-known for their optimal throughput but poor delay performance. There are several backpressure algorithms applied to mobile multi-hop networks [28], [29], [30]. Recent interesting work, BWAR [31], develops an adaptive redundancy technique for backpressure routing in DTNs. However, the idea of BWAR cannot be applied to WSN-MSs, in which packet replication should not be used due to the limited bandwidth resource of low-power wireless links. Ryu et al. [29], [30] study the backpressure flow control and routing problem in disconnected static wireless networks with mobile relays and predetermined gateways. In contrast, our work focuses on WSN-MSs where each sensor node could dynamically serve as a gateway at every opportunistic contact with a mobile sink. Our previous works [27], [32] focus on the economic and social aspects of self-interest human relays; whereas the OBC algorithm proposed in this paper seamlessly combines CA-ETX and backpressure routing to improve the delay and throughput performance of data collection in WSN-MSs.

## 3 SYSTEM MODEL

We consider a WSN-MS consisting of static sensor nodes and mobile sinks to collect sensor data using short-range communication radios (e.g., Bluetooth, Zigbee, or WiFi direct). If a sensor node is in contact with a mobile sink, it forwards its sensor data to a mobile sink directly; otherwise, it stores the data and waits for its next connection to a mobile sink or forwards its sensor data through other sensor

1. It is worth noting that although our implementation and evaluation are based on IEEE 802.15.4 radio, the principles of CA-ETX and OBC can be straightforwardly adopted for WSN-MSs with other short-range communication radios such as WiFi Direct and Bluetooth. However, the link-layer interface should be specifically designed according to the actual communication radios used.

nodes to a mobile sink in a multi-hop fashion. Topologically, the network consisting of all static sensor nodes could be either connected or disconnected. Let the sets of all sensor nodes and mobile sinks be  $N^s$  and  $N^m$  respectively, and denote  $N = N^s \cup N^m$ . The network operates in discrete time slots (e.g., seconds)  $t = 0, 1, \dots$

### 3.1 Packet Reception Probability and Expected Transmission Count

To model the unreliable wireless transmissions, we define the packet reception probability (PRR) over a wireless link  $(x, y)$ ,  $0 \leq PRR_{x,y}(t) \leq 1$ , as the probability of successfully transmitting a data packet with acknowledgement from node  $x$  to  $y$ , at slot  $t$ .  $PRR_{x,y}(t)$  is assumed to be constant within the duration of a slot, but can vary over slots and across different wireless links, due to the time-varying wireless channel quality and intermittent connectivity between static sensor nodes and mobile sinks.

From basic probability theory, we can see that the value of PRR is the reciprocal of the well-known Expected Transmission Count [24], i.e., the average number of times that a packet with acknowledgement needs to be successfully transmitted. Specifically, consider a given PRR value  $PRR_{x,y}(t)$ , the corresponding ETX value,  $ETX_{x,y}(t)$  can be computed as

$$ETX_{x,y}(t) = \frac{1}{PRR_{x,y}(t)} \geq 1. \quad (1)$$

### 3.2 Link-Layer Channel Capacity

Based on the definition of PRR, we define the logical link-layer capacity of a wireless link from node  $x \in N$  to node  $y \in N$  at time  $t$

$$c_{x,y}(t) = c^{\max} PRR_{x,y}(t) = \frac{c^{\max}}{ETX_{x,y}(t)} \geq 0, \quad (2)$$

i.e., the maximum (integer) number of sensor data packets (or bits) with acknowledgements that can be successfully transmitted from  $x$  to  $y$  at slot  $t$ . Here  $c^{\max}$  is the maximal possible  $c_{x,y}(t)$ ,  $\forall t$ , which is bounded by the data rate of the wireless radio. For instance, experimental studies show that a commonly-used IEEE 802.15.4 transceiver, CC2420 (e.g., [33]), can achieve a data rate of approximate 160 40-bytes packets per second [34] in practice. If  $c_{x,y}(t) > 0$ , we say nodes  $x$  and  $y$  are in contact at slot  $t$ ; otherwise (i.e.,  $c_{x,y}(t) = 0$ ), they are not in contact at slot  $t$ .

### 3.3 WSN-MS as a Time-Varying Weighted Graph

The states of WSN-MS at a given slot  $t \geq 0$  can be represented as a directed, complete, and time-varying weighted graph  $G(N, L, c(t))$ , where  $L$  represents the set of all possible wireless links between each pair of nodes in  $N$ , and the  $|L|$ -dimensional vector  $c(t)$  represents the vector of channel capacities over all wireless links at slot  $t$ . Since the potential sparsity of the network, most entries of  $c(t)$  are zero at any given  $t$ .

It can be seen that  $c(t)$  can characterize the time-varying channel capacities of the WSN-MS caused by both slow fading between the static sensor nodes and fast fading between sensor nodes and the mobile sinks. Therefore,  $c(t)$  also implies the stochastic process of sink mobility. We assume

that  $c(t)$  is an ergodic Markov chain that takes values on a finite (but which can be arbitrary large) discrete state space  $S$ , and it has the stationary distribution probability  $\pi_c$  for each channel capacity state  $c$ .

It has been shown that many real mobility traces exhibit a high degree of spatial regularity [35], [36], [37]. In the context of WSN-MS, this means that mobile sinks appear in some specific locations with a higher probability than others, resulting in heterogeneous opportunities of sensor nodes to meet mobile sinks. For instance, sensor nodes in shopping centers should have more opportunities to meet mobile sinks than those in park.

## 4 CONTACT-AWARE ETX

In this section, we will first present the concept of CA-ETX gradient defined in an opportunistic contact graph from a global perspective. Then we will discuss in detail how to compute the CA-ETX value for each opportunistic link at real time.

### 4.1 Opportunistic Contact Graph

We consider anycast sensor data traffic in this paper, i.e., the destination of each sensor data packet can be any mobile sink.<sup>2</sup> This allows us to use a virtual sink  $VS$  to represent all the mobile sinks in  $N^m$ . We define the opportunistic contact graph as  $G^o(N^o, L^o)$ , where  $N^o = N^s \cup \{VS\}$  represents the set of all sensor nodes and the virtual sink, and  $L^o$  represents the set of all sensor-to-sensor and sensor-VS links

$$L^o = L^s \cup \{(x, VS) : x \in N^s\},$$

where  $L^s$  is the set of all wireless links between sensor nodes.

### 4.2 Multi-Hop CA-ETX Gradient

For each opportunistic link  $(x, y)$  in  $L^o$ , we define a metric  $CA-ETX_{x,y}$  to estimate the packet transmission delay over this link. The computation of link CA-ETX values will be discussed in detail in Section 4.3. Based on the link CA-ETX values, each sensor node  $x$  can compute its node CA-ETX value,  $CA-ETX_x$  in a recursive and fully distributed way

$$CA-ETX_x = \min_{y \in N_x^o} (CA-ETX_x, CA-ETX_y + CA-ETX_{x,y}), \quad (3)$$

where

$$N_x^o = \{VS\} \cup \{y : (x, y) \in L^s\}, \quad (4)$$

represents the opportunistic contact neighbor set of  $x$ . Specifically, the CA-ETX value of the virtual sink  $CA-ETX_{VS} = 0$ . For instance, Fig. 1a shows an opportunistic contact graph example with the CA-ETX value for each link. Fig. 1b shows corresponding node CA-ETX values computed by using Eq. (3) recursively.

**Remark 1.** It can be seen that the recursive computation of node CA-ETX value over an *opportunistic contact graph* follow the same form of the computation of node ETX value over a *statically-deployed network*. This enables us to easily extend current ETX-based WSN routing protocols (e.g., CTP and RPL) to WSN-MS scenarios by using CA-ETX.

2. It is straightforward to extend our work to multi-commodity traffic models, by defining a virtual sink for each commodity.



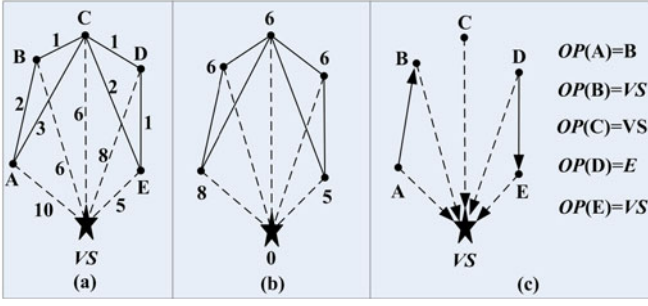


Fig. 1. An example for CA-ETX gradient in an opportunistic contact graph for WSN-MS. (a) An example opportunistic contact graph with link CA-ETX values, where solid and dashed lines represent sensor-to-sensor links and sensor-VS links, respectively, (b) corresponding node CA-ETX values, and (c) CA-ETX gradient and corresponding opportunistic shortest path routes.

For each sensor node  $x$ , let  $OSP(x, VS)$  be the shortest path from  $x$  to the virtual sink  $VS$  in the opportunistic contact graph  $G^o(N^o, L^o)$ , i.e.,  $OSP(x, VS)$  is the path with the minimal total link CA-ETX values from  $x$  to  $VS$ . It is easy to verify that the node CA-ETX value of each sensor node  $x$  represents the total link CA-ETX values of all links in  $OSP(x, VS)$

$$CA-ETX_x = \sum_{(i,j) \in OSP(x, VS)} CA-ETX_{i,j}.$$

For each sensor node  $x$ , define its opportunistic parent

$$OP(x) = \arg \min_{y \in N_x^o} (CA-ETX_y + CA-ETX_{x,y}), \quad (5)$$

where  $N_x^o$  is the opportunistic contact neighbor set of  $x$  defined in Eq. (4). For instance, Fig. 1c shows the opportunistic parents of all sensor nodes based on their node CA-ETX values.

With the CA-ETX gradient defined by the opportunistic parents of each sensor node, the opportunistic shortest path routing can be easily performed in a fully distributed way, which is summarized in Fig. 2. For instance, sensor node A in Fig. 1c will forward data to its opportunistic parent B, if it is not in contact with any mobile sink; otherwise, it will forward data to the mobile sink directly. This is different from ETX-based routing, where each node always send data to its parents.

### 4.3 Link CA-ETX Calculation

#### 4.3.1 Packet Service Time (PST)

We consider each link  $(x, y)$  in  $L^o$  as a queue with time-varying Packet Service Times  $\mu_{x,y}(t)$ ,  $t \geq 0$ , which is the time duration required for a successful packet transmission over link  $(x, y)$  at slot  $t$ .

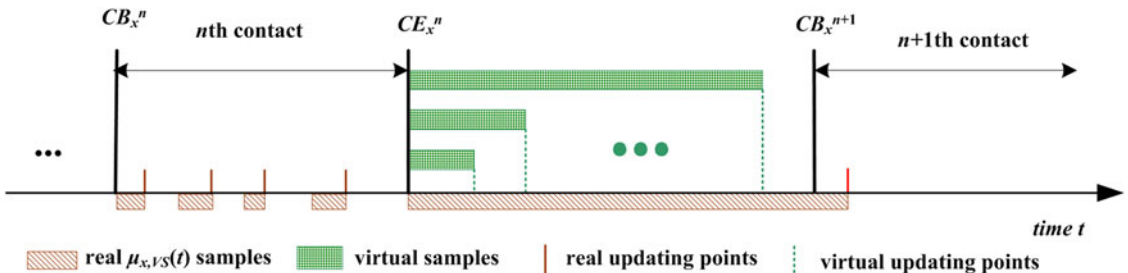


Fig. 3. An example to show how to compute and update CA-ETX value for sensor-VS links.

Each sensor node  $x$  in  $N^s$  performs the following operations:

```

1: if  $x$  is in contact with a mobile sink  $m$  then
2:    $x$  forwards its data to  $m$ ;
3: else if  $OP(x)$  is the virtual sink then
4:    $x$  waits for a sink;
5: else
6:    $x$  forwards its data to  $OP(x)$ ;
7: end if
    
```

Fig. 2. The pseudo code of distributed shortest path routing based on CA-ETX gradient.

If  $(x, y)$  is a sensor-sensor link,  $\mu_{x,y}(t)$  depends on wireless transmission rate, and can be easily computed as

$$\mu_{x,y}(t) = 1/c_{x,y}(t), \forall t \geq 0. \quad (6)$$

For instance, consider a maximum data rate of approximate 160 40-bytes packets per second achieved by CC2420 radio [34], the PST is always larger than 6.25 milliseconds. Consider the definition of logical link-layer channel capacity (2) and PST (6), we can see the classic time-average ETX value over link  $(x, y)$  is

$$ETX_{x,y} = \mathbb{E}[\mu_{x,y}(t)]c^{\max} = \mu_{x,y}c^{\max}, \quad (7)$$

where  $\mathbb{E}[\cdot]$  is the expectation operator and  $\mu_{x,y}$  is the long-term mean of  $\mu_{x,y}(t)$ .

For a sensor-VS link  $(x, VS)$ , its PST at a given slot  $t$  can be computed as

$$\mu_{x,VS}(t) = \begin{cases} 1/c_{x,m}(t), & CB_x^n \leq t \leq CE_x^n \\ CB_x^{n+1} - t + 1/c_{x,m}(CB_x^{n+1}), & \text{otherwise,} \end{cases} \quad (8)$$

where  $CB_x^n$  and  $CE_x^n$ ,  $n \geq 1$  are the first and last time slots of the  $n$ th contact between  $x$  and any mobile sink  $m$  respectively, shown in Fig. 3. According to Eq. (8), PSTs for sensor-VS links depend on the durations of each contacts, the inter-contact times (i.e., durations between each contact), as well as the wireless link quality during each contact, which cannot be reflected in existing routing metrics, such as ETX, contact probability, or inter-contact time.

#### 4.3.2 Opportunistic Link as a G/G/1 Queue

Due to the complex dynamics of the system, we do not assume any specific distribution of both the arrival process and service times over each opportunistic link  $(x, y) \in L^o$ . Therefore, we model each link in  $L^o$  as a G/G/1 queue (i.e., general distributions for packet inter-arrival and service times). From queueing theory, the average packet waiting

time  $wd_{x,y}$  in the link queue  $(x, y)$  can be approximately represented as

$$wd_{x,y} \approx \frac{\sigma_{x,y}^a + \sigma_{x,y}^s}{2(\chi_{x,y} - \mu_{x,y})}, \quad (9)$$

where  $\mu_{x,y}$  and  $\sigma_{x,y}^s$  are the standard mean and variance of PST over link  $(x, y)$  respectively, and  $\chi_{x,y}$  and  $\sigma_{x,y}^a$  are the mean and variance of packet arrival intervals respectively. From (9), we can see that  $wd_{x,y}$  is an increasing function of both  $\mu_{x,y}$  and  $\sigma_{x,y}^s$ . Therefore, we define the CA-ETX value for each link  $(x, y) \in N^o$  as

$$CA-ETX_{x,y} = (\sigma_{x,y}^s / \tilde{\sigma}_{x,y}^s) c^{\max} \mu_{x,y}, \quad (10)$$

where  $\tilde{\sigma}_{x,y}^s$  is the variance of the PST during each contact between  $x$  and  $y$ , i.e., if  $(x, y)$  is a sensor-sensor link,  $\tilde{\sigma}_{x,y}^s = \sigma_{x,y}^s$  is the variance of all PST samples; if  $(x, y)$  is a sensor-VS link,  $\tilde{\sigma}_{x,y}^s$  is the variance of all PST samples during  $CB_x^n \leq t \leq CE_x^n$ ,  $n \geq 1$ . The aim of normalizing  $\sigma_{x,y}^s$  by  $\tilde{\sigma}_{x,y}^s$  is to follow the concept of classic ETX, which further facilitates its use in current ETX-based routing protocols with the minimal modification. As a result, we have

$$CA-ETX_{x,y} = \begin{cases} c^{\max} \mu_{x,y} = ETX_{x,y} & y \in N_x^o - \{VS\} \\ (\sigma_{x,y}^s / \tilde{\sigma}_{x,y}^s) c^{\max} \mu_{x,y} & y = VS. \end{cases} \quad (11)$$

#### 4.4 Updating CA-ETX for Sensor-VS Links

For a sensor-VS link  $(x, VS)$ , PST samples can be easily estimated when  $x$  transmits each packet to any mobile sink at run time. For each new PST sample, the values of  $\mu_{x,y}$ ,  $\sigma_{x,y}^s$ , and  $\tilde{\sigma}_{x,y}^s$  can be updated based on an efficient online algorithm presented in Appendix A, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TMC.2016.2595574>.

However, when  $x$  is not in contact with any sink, PSTs of a sensor-VS link  $(x, VS)$  could be very large (e.g., in minutes or even hours), as shown in Fig. 3, while typical PSTs when  $x$  is in contact with a sink is in tens of milliseconds. As a result, simply updating  $CA-ETX_{x,VS}$  after each PST sample (e.g., the red time point in Fig. 3) would result in the CA-ETX gradient react slowly to the network dynamics. We solve this problem by using a virtual PST sample shown in Fig. 3 before a large real PST sample is obtained. This virtual sample (and also  $CA-ETX_{x,VS}$ ) is updated at time points (e.g., green time points in Fig. 3) with a small interval (e.g., current mean service time  $\mu_{x,VS}$ ). The virtual sample is abandoned when the real large service time sample is obtained.

#### 4.5 Discussion

For a sensor node  $x$ , it is easy to see that the mean of the PST  $\mu_{x,VS}$  depends on its contact probability with any mobile sink and link quality during each contact, while the variance  $\sigma_{x,y}^s$  mainly depends on the inter-contact time between  $x$  and any mobile sink. It is also not difficult to verify that the former depends on a spatial distribution of mobile sinks and deployments of sensor nodes, while the latter is greatly affected by the movement speeds of the mobile sinks.

We designed a simple proof-of-concept simulation based on Castalia [20] to illustrate above discussions. We consider

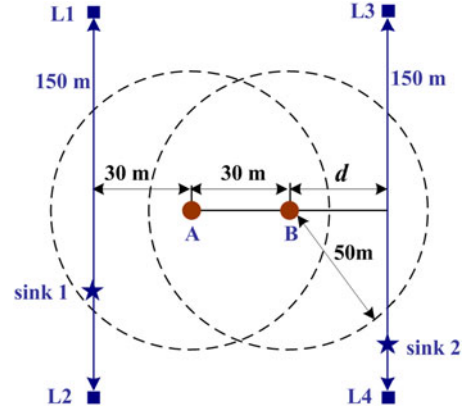


Fig. 4. An simple WSN-MS example to illustrate the impacts of sink movement (speed and trajectory) on PST behaviors and packet transmission delay.

a WSN-MS with two sensor nodes (A and B) and two mobile sinks moving repeatedly between two points (L1, L2 for sink 1, and L3, L4 for sink 2) with distances of 150 meters respectively, illustrated in Fig. 4. The transmission range of all nodes are about 50 meters (0 dbm transmission power). Each sensor node produces one 34-byte packet per second, and transmits the data to the mobile sinks using CA-ETX gradient based shortest path routing. We conduct the following studies:

##### 4.5.1 PST Illustration

We first study the impact of inter-contact time and contact duration on PST, by fixing the location of sink 1 at L1, setting  $d = 30$ , and changing the movement speed of sink 2. In this simulation, sink 1 can never communicate with the two sensor nodes, while sink 2 can communicate with B intermittently. From Figs. 5a and 5b, we can see that a change of movement speed will not affect the mean of contact probability (53.3 percent) and the mean of packet service time ( $\mu_{B,VS} = 163$  milliseconds), but will result in smaller inter-contact time and PST variance  $\sigma_{B,VS}^s$ . As a result, packet transmission delay decreases as we increase the movement speed of sink 2, shown as Fig. 5c. This verifies our queueing-theoretical analysis and discussions. In addition, Figs. 5b and 5c also demonstrate that the queueing delay (packet waiting time in queues) is the major part of end-to-end delay (several seconds) in intermittently-connected WSN-MSs, compared with data transmission delay (tens of milliseconds) caused by limited wireless channel capacity.

##### 4.5.2 Efficiency of CA-ETX

To illustrate the efficiency of CA-ETX, we used the following three metrics to measure the delay of sensor-VS links (A, VS) and (B, VS): (1)  $CA-ETX_{x,VS}$ , (2)  $\mu_{x,VS}$  (pure mean), and (3)  $\sigma_{x,VS}^s$  (pure variance). In this set of simulations, we set the speed of sink 1 as 2 m/s and  $d = 35$ . From Fig. 5d, we can see all three metrics achieve nearly same delay when sink 2 moved slower than 2 m/s. This is because the majority of sensor data packets were transmitted through route B  $\rightarrow$  A  $\rightarrow$  sink 1 (because  $d = 35 > 30$ ). When sink 2 moved faster than 2 m/s, most data were transmitted through path A  $\rightarrow$  B  $\rightarrow$  sink 2 by using pure variance and CA-ETX, while route B  $\rightarrow$  A  $\rightarrow$  sink 1 was still the major path chosen by

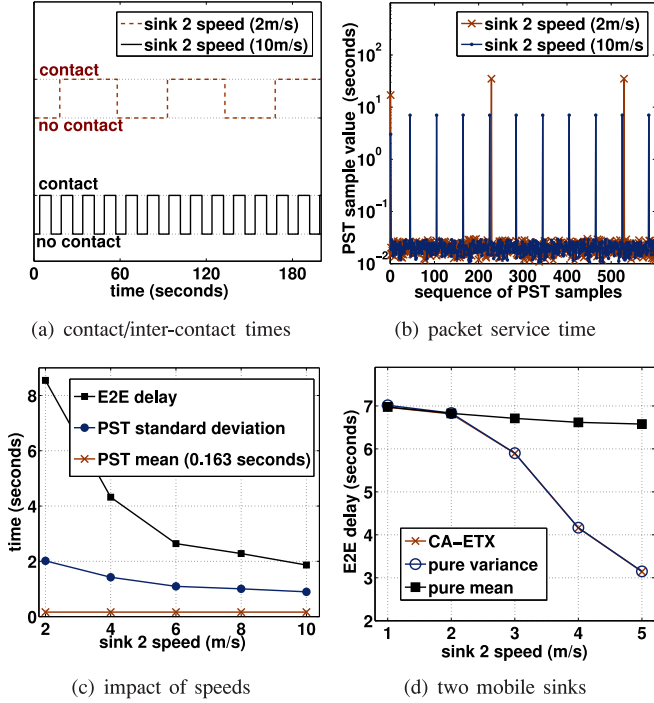


Fig. 5. An illustrative example to demonstrate the impacts of sink movement speeds and sensor deployment on PSTs and transmission delay: in (a), (b), and (c), sink 1 was fixed at L1 shown in Fig. 4 and  $d = 30$  m; in (d), sink 1 moves with a fixed speed of 2 m/s and  $d = 35$  m.

using *pure mean*. This results in a much faster delay reduction of CA-ETX and *pure variance* than that of *pure mean*, as the speed of sink 2 increases.

Since real sink mobility (e.g., walking people, bikes, and vehicles) exhibits the regular spatial behavior and heterogeneous movement speeds [35], [36], this simple illustrative example demonstrates the potential of CA-ETX in practical WSN-MSs. For instance, by using CA-ETX, data packets would be relayed via sensor nodes close to a fast moving highway rather than via nodes close to a pedestrian path when they have similar probability to contact a mobile sink, resulting in better network performance.

#### 4.5.3 CA-ETX in a Large-Scale WSN-MS

To further illustrate the efficiency of CA-ETX, we constructed another two sets of simulations using a large-scale WSN-MS with 400 sensor nodes and 50 mobile sinks in a  $1,000 \times 1,000$  m area, illustrated in Fig. 6. We use a realistic mobility model HHW [38] to simulate human mobility behaviors such as the spatial regularity (see the heterogeneous distribution of sink appearance probability shown in Fig. 6). The sink movement speeds were fixed to 5 m/s and uniformly distributed between 1 and 9 m/s in the first and second sets of simulations respectively. Other simulation settings are the same as that in Section 6.4.2. Due to the same mean of sink movement speeds (i.e., 5 m/s), the average packet service times are similar for both simulations. However, the variances of PSTs  $\sigma_{x,VS}^s$  are different, resulting in significant delay performance difference summarized in Table 1. Since CA-ETX manages to provide a fine estimation of the packet waiting time over each sensor-VS link, it achieves the best delay performance in both simulations, as shown in Table 1.

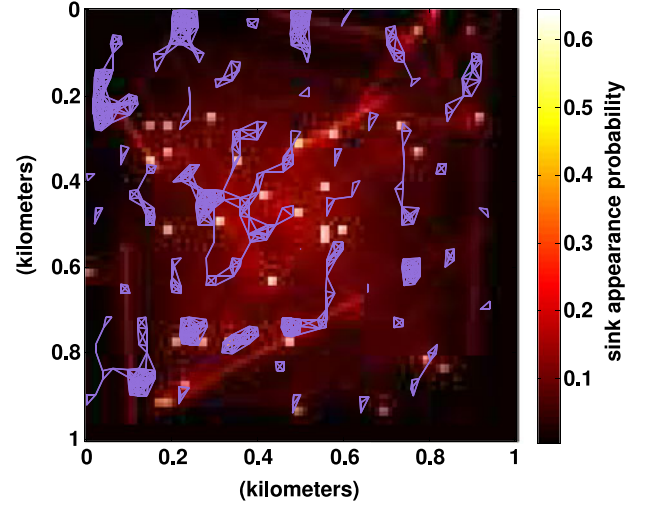


Fig. 6. Illustration of a WSN-MS (400 sensors and 50 mobile sinks) for the proof-of-concept simulation of CA-ETX, where the blue lines represents the wireless links between sensor nodes and background color visualizes sink movement, i.e., a given color presents the percentage of time a sink visited a given location (sink appearance probability).

## 5 THE OBC ALGORITHM

By forwarding sensor data through the minimal-cost routes, shortest path routing has good delay performance especially in WSN-MSs with light sensor data traffic load. However, the poor throughput performance of shortest-path routing limits its practical applications in WSN-MSs with potential large volume of sensor data traffic. Therefore, this section develops a novel throughput-optimal algorithm, OBC, by integrating CA-ETX into the backpressure algorithm for WSN-MSs. Before presenting OBC, we first model the WSN-MS as a dynamic networked queuing system.

### 5.1 Queueing Dynamics

Every sensor node  $x \in N^s$  collects data at a sensing rate of  $r_x$ .<sup>3</sup> Let  $0 \leq f_{x,y}(t) \leq c_{x,y}(t)$  represent the actual amount of sensor data transmitted over the wireless link  $(x, y)$  at slot  $t$ . Define

$$N_x(t) := \{y : c_{x,y}(t) > 0, y \in N\}, \quad (12)$$

as the set of node  $x'$  all current neighbors. Each sensor node maintains a queue (i.e., data buffer) to store the sensor data packets received from other sensor nodes and produced by itself. Let  $Q_x(t) \geq 0$  be the queue backlog (or queue length) of  $x \in N^s$  at slot  $t \geq 0$ . From slot  $t$  to  $t+1$ , queue backlog updates as follows:

$$Q_x(t+1) = |Q_x(t) - f_x^{\text{out}}(t)|_+ + r_x + f_x^{\text{in}}(t), \forall x \in N^s, \quad (13)$$

where  $f_x^{\text{in}}(t)$  and  $f_x^{\text{out}}(t)$  are the amount of total incoming and outgoing data of node  $x$  at slot  $t$ , respectively, i.e.,

$$f_x^{\text{in}}(t) = \sum_{y \in N_x(t)} f_{y,x}(t), \quad f_x^{\text{out}}(t) = \sum_{y \in N_x(t)} f_{x,y}(t),$$

3. We assume the constant sensing rate for analytical brevity. However, it is straightforward to extend our results to general ergodic sensing rates.



TABLE 1  
End-to-End Delay (in Seconds) of Opportunistic Shortest  
Path Routing with Different Link Metrics

	CA-ETX	Pure Mean	Pure Variance
simulation 1 (5 m/s)	4.96	17.12	9.82
simulation 2 (1-9 m/s)	6.73	27.25	8.35

and for any real number  $a$ , the operator  $|a|_+ = a$  if  $a > 0$ ;  $|a|_+ = 0$  otherwise. It is worth noting that the queue backlog  $Q_m(t) = 0$ , for all  $m \in N^s$ ,  $t \geq 0$ .

## 5.2 Link Rate Region

We say a set of wireless links in  $L$  are contention-free if they can be active (i.e., transmitting) simultaneously, which depends on the interference relations between them. For a channel state  $c$ , we define a  $L$ -dimensional contention-free link rate vector  $\mu(c)$ , where each entry  $l$  is the capacity  $c_l$  of the link  $l$  if link  $l$  is scheduled to transmit; otherwise, entry  $l$  is zero. The wireless links associated with the non-zero entries in  $\mu(c)$  are contention free. We further define the link rate region  $\Gamma(c(t))$  associated with channel state  $c(t)$  as the convex hull of all possible contention-free link rate vectors.

## 5.3 Network Capacity Region

We define a network capacity region  $\Lambda$  by saying that any given  $r \in \Lambda$  if there exists a joint routing and scheduling algorithm that controls  $f_{x,y}(t)$ ,  $(x, y) \in L$  at every slot  $t \geq 0$  such that

$$\overline{f_x^{out}} - r_x - \overline{f_x^{in}} = 0, \forall x \in N^s \quad (14)$$

$$f(t) \in \Gamma(c(t)), \forall t, \quad (15)$$

where  $\overline{f_x^{out}}$  and  $\overline{f_x^{in}}$  are the long-term averages of  $f_x^{in}(t)$  and  $f_x^{out}(t)$ , respectively, and  $f(t)$  is the vector of all  $f_{x,y}(t)$ ,  $(x, y) \in L$ . Constraints (14) and (15) state the flow conservation law and the link rate region constraint respectively.

## 5.4 OBC Algorithm

At each slot  $t \geq 0$ , the OBC algorithm operates as follows:

1. *Weight Calculation.* Each sensor node  $x \in N^s$  computes the weight  $w_{x,y}(t)$  for each of its current neighbors  $y \in N_x(t)$

$$w_{x,y}(t) = (Q_x(t)/\varphi_x - Q_y(t)/\varphi_y)c_{x,y}(t), \quad (16)$$

where

$$\varphi_x = \frac{1}{CA-ETX_{x,VS}},$$

is called the Gateway Quality (GQ) of sensor node  $x$ . To guarantee the stability of OBC in theory, we set deterministic lower and upper bounds for all sensor nodes  $x \in N^s$ , i.e.,  $0 < \varphi^{\min} \leq \varphi_x \leq \varphi^{\max} < \infty$ . In addition, we set  $\varphi_m$  for each mobile sink  $m$  as any fixed non-zero value between  $\varphi^{\min}$  and  $\varphi^{\max}$ , which has no impact on OBC algorithm.

2. *Scheduling.* The set of scheduled links  $F^*(t)$  is chosen as

$$F^*(t) = \arg \max_{F(t) \in \Gamma(c(t))} \sum_{(x,y) \in F(t)} w_{x,y}(t). \quad (17)$$

It is clear that  $F^*(t)$  is the set of contention-free links with the maximum aggregated weights at slot  $t$ . Current solutions to compute the optimal  $F^*(t)$  are all centralized and computational complex. However, sub-optimal  $F^*(t)$  can be computed in a fully distributed way by using the lightweight Longest Queue First (LQF) approach [39], [40], which has been shown to have a close-to-optimal performance in practice. This will be discussed in detail in Section 6.1.

3. *Routing and Forwarding.* Based on  $F^*(t)$ , each sensor node  $x \in N^s$  transmits a sensor data packet to the next one-hop node by setting  $f_{x,y}(t)$ ,  $y \in N_x(t)$  as follows:

$$f_{x,y}(t) = \begin{cases} c_{x,y}(t) & (x, y) \in F^*(t) \wedge w_{x,y}(t) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Hence, each node  $x$  chooses the next hop node  $y$  such that link  $(x, y)$  is scheduled and  $w_{x,y}(t) > 0$  (routing), then transmits  $c_{x,y}(t)$  amount of data packets to  $y$  (forwarding).

4. *Queue Update.* Each sensor node  $x$  updates its queue backlog using (13).

## 5.5 Discussions

It has been shown that classic backpressure routing can be directly used in mobile networks [6], since the long-term queue gradient can be automatically established for dynamic networks. From the theoretical point of view, this is because backpressure routing inherently minimize the per-slot Lyapunov drift in order to stabilize all queues in the network from an infinite-horizon perspective. However, the queue gradient would react much slower than the changes of network states (e.g., sink locations), resulting in long data delivery paths and significant routing loops in practice.

In order to accelerate the system reaction speed, OBC seamlessly combines mobility awareness (i.e., CA-ETX) and queue gradient of backpressure algorithm in its weight calculation (16). Here, sensor nodes with higher GQs have higher opportunities to receive more packets than those with lower GQs. Since the packet waiting time over opportunistic sensor-VS links can be estimated by CA-ETX, the packet transmission delay can be significantly reduced, compared with classic backpressure algorithm.

It can be seen that OBC is quite lightweight when a lightweight distributed scheduling algorithm is adopted. In addition, it is worth noting that OBC *does not* require predicting any future knowledge of the network, and makes routing and scheduling decisions based on current network information only. Theorem 1 below demonstrates the throughput optimality of OBC.

**Theorem 1.** *Given any traffic arrival (i.e., sensing rate vector)  $\mathbf{r}$  such that  $\mathbf{r} + \varepsilon \in \Lambda$  for any  $\varepsilon > 0$ , the OBC algorithm (with optimal scheduler) can stabilize all queues, i.e.,*

$$\limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K \sum_{x \in N} \mathbb{E}[Q_x(t)] < \infty$$

TABLE 2  
Implementation Details

Protocol	CA-ETX (CTP)	CA-ETX (RPL)	OBC (testbed)
Platform	TinyOS	Contiki OS	TinyOS
Code Changed	35 lines	68 lines	-
Total RAM (kB)	3.1	9.8	2.7
Total ROM (kB)	30.5	42.5	26.9

**Proof.** The proof of Theorem 1 is presented in Appendix B, available in the online supplemental material.  $\square$

## 6 EVALUATION

In this section, we will construct extensive real-world experiments and simulations to evaluate the proposed CA-ETX metric and OBC algorithm. Our evaluation are based on two popular WSN and IoT operating systems, TinyOS [17] and Contiki OS [18]; and a realistic WSN simulator Castalia [20].

We applied the CA-ETX metric to the defacto TinyOS routing standard CTP [5] and the IETF IPv6 routing protocol RPL (in Contiki OS) [16], [41], and implemented the OBC algorithm in both TinyOS and the Castalia simulator, as shown in Table 2. In addition, we adopted the following implementation approaches in our evaluations.

**Sink Discovery and Neighbor Management.** In our evaluation, each mobile sink declares its presence to its current nearby sensor nodes by periodically broadcasting one-hop beacons. We set the beacon broadcasting interval of mobile sink  $T_{\text{sink}}$  as 1 second and 250 milliseconds for testbed experiments and simulations respectively, as shown in Table 3. Each sensor node also broadcasts a beacon per second to inform its local information (e.g., queue backlog) to its neighbors. In addition, each node updates its neighbor table every  $T_{\text{neighbor}} = 50$  milliseconds. To reduce the control packets, we implemented the overhearing (or snooping) mechanism (e.g., [6]). Here, each data packet or acknowledgment message includes the sender's local information in a packet header filed. A node (sensor node or mobile sink) does not need to broadcast a beacon in a broadcasting interval, if it has already sent a data packet or replied an acknowledgment in the same broadcasting interval. It can be seen that our implementation achieves both precise and lightweight sink discovery and neighbor table updating.

**Distributed Scheduling.** The optimal solution to the scheduling problem (17) is centralized and NP-hard for practical wireless networks with general interference relations (e.g., [42]), which is therefore intractable in practical WSN-MSs. To solve this problem, we implemented a fully distributed suboptimal scheduler, the greedy *Longest Queue First*, which can achieve a near-optimal performance in practical wireless networks [40], [43], [44]. Due to the page limit, we do not describe the details of distributed LQF implementation, which can be found in [40].

**CSMA-Based WSN-MSs.** Consider the discrete time slot modeling in our system, the *distributed OBC with LQF scheduler* can be directly used in synchronized TDMA networks. However, Since most current wireless devices adopt CSMA-based radios, our evaluation used an efficient technique proposed in [45], [46] to implement OBC with LQF in a fully asynchronous way: If a link  $(x, y)$  is scheduled to transmit (decision made by OBC with LQF), node  $x$  will reduce its CSMA backoff window size to aggressively access the channel; otherwise,  $x$  accesses the channel with normal backoff window size.

### 6.1 Implementation

### 6.2 Evaluation Summary and Parameter Setting

Throughout the evaluation, we collected the following four metrics to measure the performance of all protocols.

- *End-to-End Delay.* The time taken for every packet from source to destination.
- *Queue Backlog.* The number of data packets in each node's data buffer, which indicates the storage efficiency.
- *Communication Overhead.* The number of transmitted and received packets (including all data and control packets) per node per second. This performance metric can measure the efficiency of routing algorithm in terms of energy and bandwidth consumptions.
- *Packet Loss Rate.* The percentage of lost data packets, indicating the reliability performance. Here, the remaining packets at the end of each evaluation was not considered as packet losses.

Table 3 summarizes the parameter settings of our evaluation, which will be discussed in detail in specific sections below.

TABLE 3  
Summary of Evaluation Parameter Settings

Evaluation	CA-ETX Evaluation (Subsection 6.3)		OBC Evaluation (Subsection 6.4)			
	Testbed	Simulation	Testbed	Simulation		
Method	CTP(CA-ETX vs ETX)	RPL(CA-ETX vs ETX)	OBC&BCP	OBC	BP	MG-IP
Protocols	TinyOS/MicaZ	Contiki/Cooja	TinyOS/MicaZ	Castalia	Castalia	Castalia
Platform	TinyOS/MicaZ	Contiki/Cooja	TinyOS/MicaZ	Castalia	Castalia	Castalia
MAC Layer	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4	CSMA+LQF	CSMA+LQF	CSMA
TX power	-25 dbm	0 dbm	-25 dbm	0 dbm	0 dbm	0 dbm
Mobility	Real Mobility	HHW Model	Real Mobility	HHW Model	HHW Model	HHW Model
Prediction	no	no	no	no	no	more than 95% accuracy
Packet Size	34 Bytes	40 Bytes	34 Bytes	34 Bytes		
Retransmission	10 times	10 times	10 times	10 times		
Scale	20 sensors+2 sinks	200 sensors+10 sinks	20 sensors+2 sinks	up to 1000 sensors+20 sinks		
$T_{\text{sink}}$	1 second	250 milliseconds	1 second	250 milliseconds		
$T_{\text{neighbor}}$	50 milliseconds	50 milliseconds	50 milliseconds	50 milliseconds		
Buffer size	20 packets	40 packets	20 packets	300 packets		





Fig. 7. Sensor deployments for the ETX and CA-ETX (with CTP) experiments.

### 6.3 Evaluation of CA-ETX

In this section, we demonstrate how to use the CA-ETX metric to extend current WSN routing protocols to WSN-MSs. Specifically, we first applied CA-ETX to CTP [5], and evaluate the performance improvement through a testbed experiment using TinyOS based MicaZ motes. Then, we constructed a simulation using Cooja, the simulator of Contiki OS [18], to demonstrate the performance gain of RPL [16] by using CA-ETX.

#### 6.3.1 CA-ETX with CTP

We constructed experiments using MicaZ motes to evaluate the practical performance of applying CA-ETX in CTP [5]. As shown in Table 2, only 35 lines of nesC code is used to implement CA-ETX in CTP. We compared the performance of CTP with ETX and CTP with CA-ETX by using the following methods:

As shown in Figs. 7 and 8, two WSN-MSs were concurrently deployed for the ETX and CA-ETX experiments respectively in London Hyde Park. Each WSN-MSs consisted of 20 sensor nodes (i.e., the blue cycles in Fig. 8a), which were deployed in a grid topology with a cell size of two meters. The mobile sinks were held by two researchers respectively (each researcher carries two sinks for ETX and CA-ETX respectively). The experiment lasted for 20 minutes

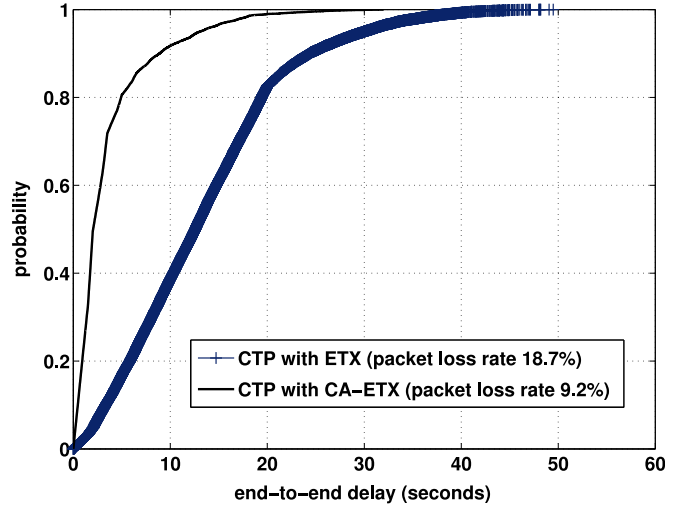


Fig. 9. Packet loss rate and CDF of end-to-end delay: CTP with ETX and CA-ETX.

during which the two researchers (mobile sinks) roamed around the deployment area, simulating both high probabilities of visiting some hot points (i.e., red circles and stars in Fig. 8a) and other low probability locations. Therefore, the two WSN-MSs had the exactly same deployments and sink mobility. In order to avoid inter-interference between the two WSN-MSs, they were operated on two orthogonal channels of the CC2420 radio (channels 13 and 26) respectively. In addition, neither channel experienced interference by other external 2.4 GHz wireless transmissions during the experiment, as no WiFi signal was detectable around the deployment area.

Fig. 8 illustrates above experiment scenarios, where the diameter of each blue cycle in Fig. 8a is linearly proportional to the percentage of time that the corresponding sensor node was in contact with a mobile sink; the width of each blue line is linearly proportional to the percentage of time that corresponding pair of sensors were connected as neighbors; and Fig. 8b illustrates the time sequences of neighbor numbers of the two mobile sinks.<sup>4</sup>

In this experiment, the packet size, node transmission power, and sensing rate, were set as 34 bytes, -25 dBm (results in around 2-3 meter transmission range), and one packet per five seconds for each sensor node respectively. Fig. 9 shows the Cumulative distribution function (CDF) of the end-to-end packet transmission delays. It can be seen that more than 90 percent of packets were transmitted within 10 seconds by using CTP with CA-ETX, while only 40 percent packets in the CTP with ETX experiment achieved this performance. The average delay of the CA-ETX experiment is 3.6 seconds, which is 73 percent less than that of ETX experiment (i.e., 13.4 seconds).

As shown in Fig. 9 and Table 4, CTP with CA-ETX also achieves smaller packet loss rate, lighter average storage overhead (queue length) and similar communication overhead (Tx/Rx rate), compared with CTP with ETX. This experiment shows that although the real-time routing

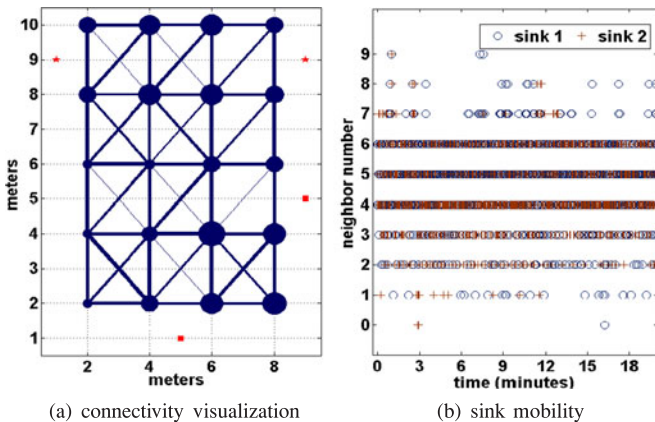


Fig. 8. Visualization of the ETX and CA-ETX (with CTP) testbed experiment settings: (a) Sensor deployment topology and mobility hot spots and (b) the sequences of neighbor numbers of the two sinks.

4. Since the collected illustration results are almost same for the two WSN-MSs with orthogonal channels, we plot the CA-ETX experiment results in Fig. 8 for brevity.

TABLE 4  
End-to-End Delay (in Seconds) of Opportunistic Shortest Path Routing with Different Link Metrics

	queue backlog	Tx/Rx Rate
CTP with ETX	10.32 packets	5.63 packets/node/second
CTP with CA-ETX	7.83 packets	5.77 packets/node/second

protocol CTP is originally designed for static WSNs, it has a great potential to be extended to support delay-tolerant data traffic in the intermittently-connected WSN-MSs, by using CA-ETX.

### 6.3.2 CA-ETX with RPL

In this section, we evaluated the performance of RPL with CA-ETX by using Cooja, the simulator of Contiki OS. As shown in Table 2, we modified 68 lines of code to implement CA-ETX and an enhanced loop detection scheme to RPL. We also changed some codes in Contiki's IPv6 stack to support node mobility and delay-tolerant applications, including implementing a queue at the network layer, faster neighbour table updates, and broadcasting neighbour solicitation messages more frequently.

In this set of simulations, we consider a multi-hop WSN-MS consisting of 200 randomly-deployed sensor nodes and 10 mobile sinks in a  $500 \times 500$  m simulation area. Each sensor node generated one UDP packet (40 bytes) per 40 seconds. We use a realistic mobility model, the Heterogeneous Human Walk (HHK) model [38] to simulate sink mobility. The sink movement speeds were randomly set between 1 and 9 m/s respectively. Each simulation lasted for 2,000 seconds. Fig. 10 shows the CDF of end-to-end delay for the modified RPL/IPv6 with ETX and CA-ETX respectively.

It can be seen from Fig. 10 that using ETX and CA-ETX result in similar packet loss rates of RPL, and the transmission delay of around 20 percent data packets are similar (less than 1 seconds) when using the two metrics. These 20 percent data packets could be either sent from sensor nodes directly to the sink within one-hop, or through temporally existing multi-hop paths. In comparison to ETX, however, CA-ETX significantly reduces the delay of data packets

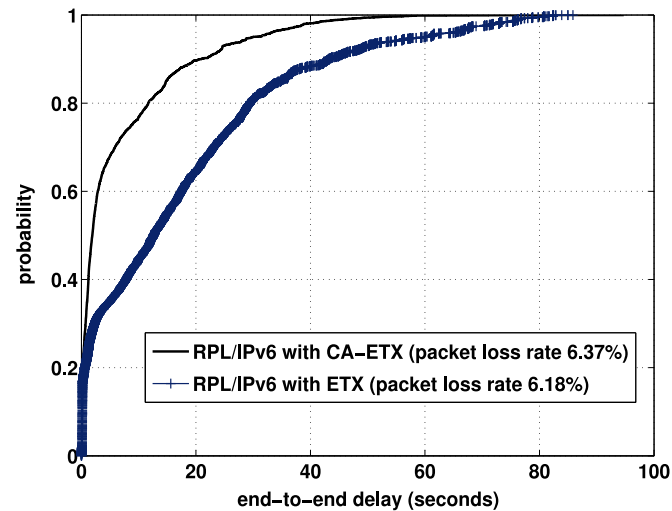


Fig. 10. Packet loss rate and CDF of end-to-end delay: Modified RPL/IPv6 with ETX and CA-ETX.

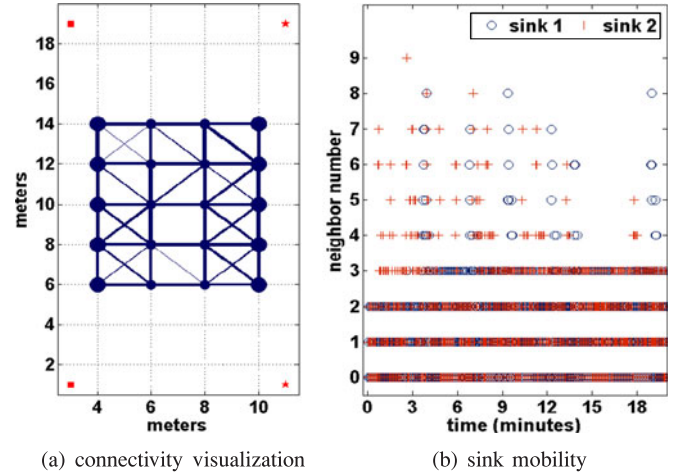


Fig. 11. Visualization of the BCP and OBC testbed experiment settings: (a) Sensor deployment topology and mobility hot spots and (b) the number of neighbors for the two sinks.

(around 80 percent) that were transmitted through opportunistic multi-hop paths. As a result, the overall average of end-to-end delay of RPL with ETX (19.92 seconds in average) is approximately three times larger than that of CA-ETX (6.63 seconds in average).

## 6.4 Evaluation of OBC

### 6.4.1 Testbed Experiments

In this section, we compared OBC with a practical backpressure-based WSN routing protocol BCP [6], using real-world experiment with TinyOS-based MicaZ motes. Since BCP is a pure backpressure routing protocol without scheduling, we only compared the routing part of OBC with BCP for fairness. The method of this experiment was the same as that of the CA-ETX experiment in Section 6.3.1 but sensor layouts and sink mobility were different, which are illustrated in Fig. 11. In addition, sensing rates of OBC and BCP were set as one packet per two seconds, while all other parameter settings were the same as that in the CA-ETX experiment.

As shown in Fig. 12, OBC significantly outperforms BCP in end-to-end delay, storage overhead, and communication overhead, while achieving similar packet loss rates as BCP. When a sensor node  $x$  transmits data packets to a mobile sink in the WSN-MS running BCP, its queue length reduces, resulting in a queuing gradient towards  $x$ . However, the mobile sink may disconnect from  $x$  before the convergence of such a gradient, resulting in severe routing loops. Such time-varying queue gradients caused by sink mobility aggravate the hop count performance of pure backpressure routing which is known to perform poorly already in static networks (e.g., [47]). In OBC, however, data packets are continuously forwarded to sensor nodes with low node CA-ETX values, which significantly reduces convergent time therefore mitigates routing loops (i.e., CA-ETX gradient helps the convergence of queue gradient). This results in much less unnecessary data transmissions (46 percent less average TX/RX rates), as shown in Fig. 12c.

Furthermore, the network capacity resource (i.e., opportunistic contacts between sensors and mobile sinks)

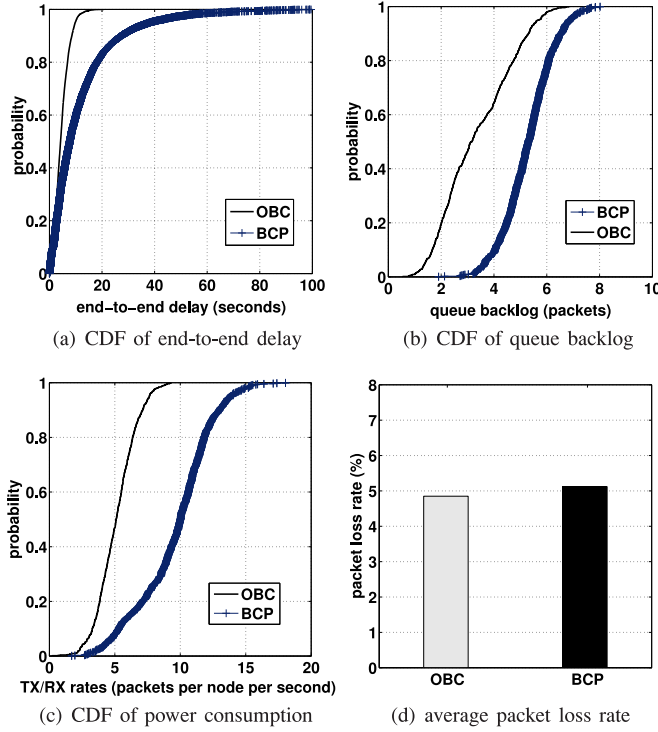


Fig. 12. OBC experiment results.

is better utilized by OBC, compared with BCP. This is because that a sensor node cannot transmit data to any nearby mobile sink during the slots when its queue is empty, while OBC minimizes the number of such slots by ensuring that better gateways (sensors with lower CA-ETX value) have high probability to receive more data. However, BCP treats all sensor nodes homogeneously and inherently tries to balance all queues in the network. Therefore, the probability of empty queues in good gateways is much higher, resulting in a waste of sensor-sink contact opportunities.

Due to the better network capacity resource usage and less data forwarding hops (less routing loops), OBC achieves much less network congestion and therefore much smaller queue backlog (59.1 percent in average) and less end-to-end delay (38.2 percent in average) than BCP, shown in Figs. 12b and 12a respectively.

#### 6.4.2 Simulation Study

We constructed extensive simulations to further evaluate the performance of OBC, in terms of throughput, adaptability to sink movement, and scalability.

*Simulation Settings.* We compared OBC with a state-of-the-art protocol in WSN-MS that is based on mobility graph and information potentials (MG-IP) [15], and the classic backpressure routing/scheduling algorithm (BP). As shown in Table 3, all the three protocols were implemented on the top of Castalia CSMA link layer. We implemented LQF and the back-off window adjustment technique [45] for the distributed scheduling of OBC and BP.

It is worth noting that OBC and BP *do not* require any knowledge of the future system states, but MG-IP requires to forecast future sink mobility. In our simulations, we set the mobility prediction accuracy of a mobile sink  $m$ ,  $PAC_m$

as a decreasing function of  $m$ 's speed  $spd_m$  for MG-IP

$$PAC_m = \begin{cases} (100 - spd_m/4)\%, & \text{if } spd_m \leq 20 \text{ (m/s)} \\ 95\% & \text{otherwise,} \end{cases} \quad (18)$$

which is much higher than the accuracy of prediction method reported in [15]. The aim of this setting is to show the limitations of prediction-based WSN-MS protocols, and to demonstrate that OBC can significantly outperform MG-IP even when its prediction accuracy is nearly perfect. Every simulation was run five times to obtain the average result.

*Impacts of Traffic Loads.* Figs. 13a, 13b, 13c, and 13d show the performance of the three algorithms with different sensing rate in a WSN-MS consisting of 200 sensor nodes and four mobile sinks, in a  $600 \times 600$  m area. The average sink mobility speed is 5 m/s. As shown in Fig. 13a, MG-IP and BP show opposite trends when the network data traffic changes. In general, end-to-end delay mainly depends on two factors: queue backlog (i.e., Little's Theorem) and routing path length. Since the routing decision of the mobility-aware MG-IP is independent of network traffic, it suffers from larger delay in simulations with heavier network traffics, caused by monotonically increased queue length shown in Fig. 13b. However, BP makes routing decision based on queue backlogs. As network traffic load increases, its delay decreases. This is because the reduced routing loop is a more dominant factor than the increased queue backlog.<sup>5</sup> Figs. 13b and 13c respectively show that purely queue-aware BP achieves better queue length but worse communication overhead than the purely mobility-aware MG-IP. By combining mobility and queue awareness, OBC achieves the best delay performance in almost all these simulations. Finally, the packet drops of MG-IP are caused by both wireless contentions and imperfect mobility predictions, leading to worse reliability performance than BP and OBC (no prediction requirement), as shown in Fig. 13d.

*Impacts of Sink Movement Speed.* Figs. 13e, 13f, 13g, and 13h show the performance of the three algorithms with different sink movement speeds. In this set of simulations, the WSN-MS also consists of 200 sensor nodes and four mobile sinks in a  $600 \times 600$  m area. As the sinks moved faster, the variance of packet service times over sensor-VS links decreases, as we discussed in Section 4.5. This results in faster packet transmission, lighter network congestion, and less routing loops, for all three routing protocols, shown in Figs. 13e and 13f respectively. In addition, MG-IP exhibits a large packet loss rate in simulations with high-speed mobile sinks, caused by non-ignorable mobility prediction errors, while BP and OBC are relatively insensitive to sink movement speed in terms of transmission reliability. Furthermore, that reliability performance of MG-IP would be significantly degraded in practical WSN-MSs, where high prediction accuracy (e.g., more than 95 percent in our simulations) is impossible to achieve.

5. Similar observations are also reported in static networks (e.g., [47]). Typically, when we increase the network traffic load, the delay of multipath backpressure routing first increases (due to the domination of reduced hop counts), and then decreases (caused by the domination of increased queue backlog). Our results belongs to the first phase.



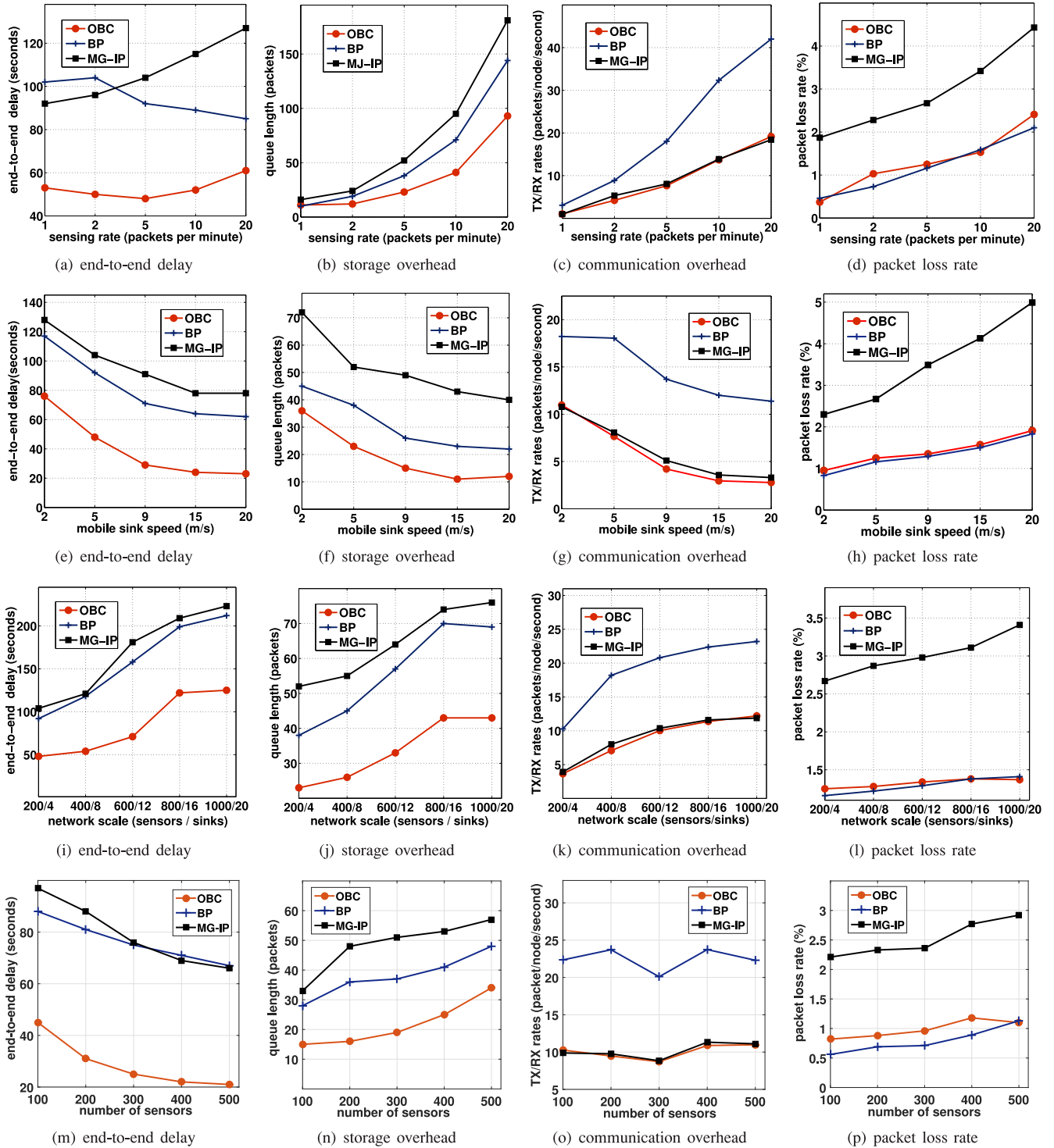


Fig. 13. Simulation results of OBC, BP, and MG-IP: (a)–(d) with different sensing rates (average sink speed is 5 m/s, each simulation has 200 sensors and 4 sinks in a  $600 \times 600$  m area.); (e)–(h) with different sink speed (sensing rate is two packets per minute); (i)–(l) with different scale WSN-MSs (average sink speed and sensing rate are 5 m/s and two packets per minute, respectively, the average node density of different scale networks remained the same, by adjusting the size of corresponding simulation areas; and (m)–(p) with different network density (average sink speed and sensing rate are 5 m/s and two packets per minute, respectively. Each simulation has 10 sinks and in a  $600 \times 600$  m area.).

**Scalability Study.** Since all three protocols only require local information (i.e., queue backlogs, CA-ETX, and information potentials) to make routing and data forwarding decisions, their control complexities are  $O(1)$  with respect to the network size, which demonstrates their potential to scale to large WSNs-MSs. However, the simulation results shown in Figs. 13i, 13j, 13k, and 13l demonstrate that

the performance of all three protocols generally decreases as the network scale increases, but the performance degradation speed of OBC is the slowest. It can also be seen that OBC outperforms the other two protocols in both small-scale and large-scale WSN-MSs. Besides network size, routing protocols in WSN-MSs are also expected to scale with respect to the number of sinks. Both OBC and BP only

require to maintain one data queue for anycast data traffic, and adopt lightweight mobility scheme (i.e., one-hop beacon and CA-ETX) without maintaining any information of a specific moving sink. Therefore, they are relatively insensitive to sink population. In contrast, MG-IP needs to store  $n$  information potential values for  $n$  mobile sinks respectively.

**Impacts of Network Density.** Figs. 13m, 13n, 13o, and 13p show the performance of the three algorithms in networks with different density. As the number of sensor node increases, the network become denser and therefore more connected. We can see that when network becomes denser, all three protocols perform significantly better in terms of end-to-end delay, while slightly worse in terms of storage overhead and packet loss rate. It can be seen that although OBC has the similar performance as MG-IP in communication overhead and BP in reliability, it significantly outperforms other two protocols in terms of delay and storage overhead for all simulations.

## 7 CONCLUSION

In this paper, we study how to improve the delay and throughput performance for delay-tolerant data collection applications in Wireless Sensor Networks with Mobile Sinks (WSN-MSs). We propose a novel routing metric, CA-ETX, based on queueing analysis theory to estimate the packet transmission delay over opportunistic links. By implementing CA-ETX in CTP and RPL, we demonstrate that current ETX-based routing protocols for WSN with static sinks can be easily applied to WSN-MSs by using CA-ETX. We also introduce a throughput-optimal data collection scheme, OBC, by integrating CA-ETX into the Lyapunov optimization framework. In contrast to current WSN-MS schemes, OBC does not require any mobility prediction and performs well in large-scale sensor networks with multiple fast moving sinks. Test-bed experiments and extensive simulations demonstrate that OBC significantly outperforms state-of-the-art approaches. Interesting future directions are the extension of our approach to the duty-cycled WSN-MSs.

## ACKNOWLEDGMENTS

This work is sponsored by Intel Corporation, China "1000 Young Talents Program", and "Young Talent Support Plan" at Xi'an Jiaotong University. The authors thank the associate editor and anonymous reviewers for their helpful and insightful comments and suggestions, which significantly contribute to improving the quality of the paper.

## REFERENCES

- [1] U. Park and J. Heidemann, "Data muling with mobile phones for sensor networks," in *Proc. 9th ACM Conf. Embedded Networked Sensor Syst.*, 2011, pp. 162–175.
- [2] X. Wu, K. N. Brown, and C. J. Sreenan, "Analysis of smartphone user mobility traces for opportunistic data collection in wireless sensor networks," *Pervasive Mobile Comput.*, vol. 9, pp. 881–891, 2013.
- [3] P.-Y. Chen, S. Yang, and J. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3832–3842, Jun. 2015.
- [4] S. Yang and J. McCann, "Distributed optimal lexicographic maximin rate allocation in solar-powered wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 11, no. 1, 2014, Art. no. 9.
- [5] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. 7th ACM Conf. Embedded Networked Sensor Syst.*, 2009, pp. 1–14.
- [6] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 279–290.
- [7] S. Yang, X. Yang, J. McCann, T. Zhang, G. Liu, and Z. Liu, "Distributed networking in autonomic solar powered wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 750–761, Dec. 2013.
- [8] W. Du, Z. Li, J. C. Liando, and M. Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," in *Proc. 12th ACM Conf. Embedded Networked Sensor Syst.*, 2014, pp. 134–147.
- [9] R. Wohlers, N. Trigoni, R. Zhang, and S. Ellwood, "TwinRoute: Energy-efficient data collection in fixed sensor networks with mobile sinks," in *Proc. 10th Int. Conf. Mobile Data Manage. Syst. Services Middleware*, 2009, pp. 192–201.
- [10] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas, "Data stashing: Energy-efficient information delivery to mobile sinks through trajectory prediction," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 291–302.
- [11] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Commun.*, vol. 19, no. 3, pp. 96–104, Jun. 2012.
- [12] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device to device communications with WiFi direct: Overview and experimentation," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 96–104, Jun. 2013.
- [13] U. Adeel, S. Yang, and J. A. McCann, "Self-optimizing citizen-centric mobile urban sensing systems," in *Proc. 11th Int. Conf. Autonomic Comput.*, 2014, pp. 161–167.
- [14] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2011, pp. 2246–2254.
- [15] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas, "Predictive QoS routing to mobile sinks in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 109–120.
- [16] "RPL: IPv6 routing protocol for low-power and lossy networks," IETF RFC 6550, 2012.
- [17] TinyOS: [Online]. Available: <http://www.tinyos.net/>
- [18] Contiki OS: [Online]. Available: <http://www.contiki-os.org/>
- [19] L. Georgiadis, M. Neely, M. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*. Breda, Netherlands: Now Publishers, 2006.
- [20] Castalia Simulator: [Online]. Available: <https://castalia.forge.nicta.com.au/index.php/en/>
- [21] H. Kim, T. Abdelzaher, and W. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Networked Sensor Syst.*, 2003, pp. 193–204.
- [22] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. Guibas, "Composable information gradients in wireless sensor networks," in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw.*, 2008, pp. 121–132.
- [23] J. Lee, B. Kusy, T. Azim, B. Shihada, and P. Levis, "Whirlpool routing for mobility," in *Proc. 11th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2010, vol. 10, pp. 131–140.
- [24] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 134–146.
- [25] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–13.
- [26] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3119–3127.
- [27] S. Yang, U. Adeel, and J. A. McCann, "Selfish mules: Social profit maximization in sparse sensor networks using rationally-selfish human relays," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1124–1134, Jun. 2013.
- [28] A. Dvir and A. Vasilakos, "Backpressure-based routing protocol for DTNs," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 405–406, 2010.
- [29] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, "Back-pressure routing and rate control for ICNs," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 365–376.
- [30] J. Ryu, L. Ying, and S. Shakkottai, "Back-pressure routing for intermittently connected networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

- [31] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Backpressure with adaptive redundancy (BWAR)," in *Proc. IEEE INFOCOM*, 2012, pp. 2300–2308.
- [32] S. Yang, U. Adeel, and J. McCann, "Backpressure meets taxes: Faithful data collection in stochastic mobile phone sensing systems," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1490–1498.
- [33] MicaZ Mote: [Online]. Available: [http://www.memsic.com/user-files/files/Datasheets/WSN/micaz\\_datasheet-t.pdf](http://www.memsic.com/user-files/files/Datasheets/WSN/micaz_datasheet-t.pdf)
- [34] A. Sridharan and B. Krishnamachari, "Explicit and precise rate control for wireless sensor networks," in *Proc. 7th ACM Conf. Embedded Networked Sensor Syst.*, 2009, pp. 29–42.
- [35] M. Gonzalez, C. Hidalgo, and A. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [36] X. Zhang, J. Kurose, B. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing," in *Proc. 13th Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2007, pp. 195–206.
- [37] X. Yang, X. Fan, W. Yu, X. Fu, and S. Yang, "HLLS: A history information based light location service for manets," *Comput. Netw.*, vol. 56, no. 2, pp. 731–744, 2012.
- [38] S. Yang, X. Yang, C. Zhang, and E. Spyrou, "Using social network theory for modeling human mobility," *IEEE Netw.*, vol. 24, no. 5, pp. 6–13, Sep./Oct. 2010.
- [39] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks: A partitioning approach," in *Proc. 12th Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2006, pp. 26–37.
- [40] L. Chen, S. Low, M. Chiang, and J. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–13.
- [41] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, and K. K. Leung, "A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.
- [42] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, 2006, pp. 227–238.
- [43] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1132–1145, Aug. 2009.
- [44] S. Yang, Z. Sheng, J. A. McCann, and K. K. Leung, "Distributed stochastic cross-layer optimization for multi-hop wireless networks with cooperative communications," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2269–2282, Oct. 2014.
- [45] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," in *Proc. 27th Conf. Comput. Commun.*, 2008, pp. 619–627.
- [46] A. Sridharan, S. Moeller, B. Krishnamachari, and M. Hsieh, "Implementing backpressure-based rate control in wireless networks," in *Proc. Inf. Theory Appl. Workshop*, 2009, pp. 341–345.
- [47] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun. 2011.
- [48] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Algorithms for computing the sample variance: Analysis and recommendations," *Amer. Statistician*, vol. AS-37, no. 3, pp. 242–247, 1983.



**Shusen Yang** received the PhD degree in computing from Imperial College London, in 2014. He is currently a professor in the Institute of Information and System Science, Xi'an Jiaotong University (XJTU). Before joining XJTU, he worked as a lecturer (assistant professor) at the University of Liverpool from 2015 to 2016, and a research associate in Intel Collaborative Research Institute (ICRI) for sustainable connected cities from 2013 to 2014. His research interests include mobile and wireless networks, networks with human in the loop, and data-driven networked systems. He is a recipient of China "1000 Young Talents Program", and an honorary research fellow at Imperial College London. He is a member of ACM, the IEEE, and IEEE Communication Society.



**Usman Adeel** received the BS degree from FAST-National University and the MS degree from GIK institute, both in Pakistan. He received the PhD degree in computing from Imperial College London, United Kingdom, in 2015. He is currently a researcher at Intel Labs Europe. His research interests include in device-to-device communications, mobile crowdsourcing, and opportunistic networking.



**Yad Tahir** received the BSc degree with honors in computer science from the University of Sulaimani (UoS), Iraq, in 2008, and the MSc degree in software engineering with distinction from Heriot-Watt University, United Kingdom, in 2010. Currently, he is working toward the PhD degree in computing at Imperial College London, United Kingdom. His research interests include resource managements, network control and optimizations, internet of things, and software and system engineering.



**Julie A. McCann** is a professor in computer systems at Imperial College. Her research centers on highly decentralized and self-organizing scalable algorithms for spatial computing systems, e.g., wireless sensing networks. She leads both the Adaptive Embedded Systems Engineering Research Group and the Intel Collaborative Research Institute for Sustainable Cities, and is currently working with NEC and others on substantive smart city projects. She has received significant funding through bodies such as the UK's EPSRC, TSB, and NERC as well as various international funds, and is an elected peer for the EPSRC. She has actively served on, and chaired, many conference committees and is currently an associate editor for the *ACM Transactions on Autonomous and Adaptive Systems*. She is a fellow of the BCS.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).