Ad Hoc Networks 11 (2013) 1150–1164

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Opportunistic data collection for disconnected wireless sensor networks by mobile mules

Yu-Chee Tseng *, Fang-Jing Wu, Wan-Ting Lai

Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan

ARTICLE INFO

Article history:
Received 17 November 2011
Received in revised form 28 December 2012
Accepted 2 January 2013
Available online 11 January 2013

Keywords:
Delay-tolerant networks
Distributed computing
Distributed storage
Mules
Protocols
Wireless communications
Wireless sensor networks

ABSTRACT

This paper considers a field with a number of isolated wireless sensor networks served by some mobile mules and base stations (BSs). Sensing data needs to be carried by mobile mules to BSs via *opportunistic contact* between them. Also, such contact may not be frequent. Thus there are four types of communications in this environment: (i) inter-node communications within a WSN, (ii) opportunistic WSN-to-mule communications, (iii) opportunistic mule-to-mule communications, and (iv) opportunistic mule-to-BS communications. In such disconnected WSNs, since sensors' memory spaces are limited and data collection from isolated WSNs to mules and then to BSs relies on opportunistic communications in the sense that contact between these entities is occasional, *storing* and *collecting* higher-priority data is necessary. Therefore, there are two critical issues to be addressed: the *data storage management* in each isolated WSN and *opportunistic data collection* between these entities. We address the storage management problem by modeling the limited memory spaces of a WSN's sensor nodes as a distributed storage system. Assuming that there is a sink in the WSN that will be visited by *mobile mules* occasionally, we address three issues: (i) how to buffer sensory data to reduce data loss due to a shortage of storage spaces, (ii) if dropping of data is inevitable, how to avoid higher-priority data from being dropped, and (iii) how to manage the data nearby the sink to facilitate the downloading jobs of mules when the downloading time is unpredictable. We propose a *Distributed Storage Management* (*DSM*) strategy based on a novel shuffling mechanism similar to heap sort. It allows nodes to exchange sensory data with neighbors efficiently in a distributed manner. For the opportunistic data collection problem, based on a utility model, we then develop an *Opportunistic Data Exchange* (*ODE*) strategy to guide two mules to exchange data that would lead to a higher reward. To the best of our knowledge, this is the first work addressing distributed storage strategy for isolated WSNs with opportunistic communications using mobile mules. We conduct extensive simulations to investigate the merit of DSM and ODE. The simulation results indicate that the level of data importance collected by our DSM is very close to a global optimization and our ODE could facilitate delivery of important data to BSs through mules. We also implement these strategies in a real sensor platform, which demonstrates that the simple and lightweight protocols can achieve our goals.

1. Introduction

Wireless sensor networks (WSNs) have gained much attention recently [1–3]. A WSN is composed of a large number of nodes, each of which has multiple onboard sensors to collect environment data. Nodes can communicate with each other through their wireless interfaces. WSNs have many applications such as military safety, health care, environmental monitoring, surveillance systems, and social networks [4–8].

We are interested in the data collection issue for *disconnected* WSNs [9] that are separated into multiple isolated groups and do not have network connectivity to outside

* Corresponding author. Tel.: +886 3 513 1218; fax: +886 3 572 4176.
E-mail address: yctseng@cs.nctu.edu.tw (Y.-C. Tseng).

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.01.001

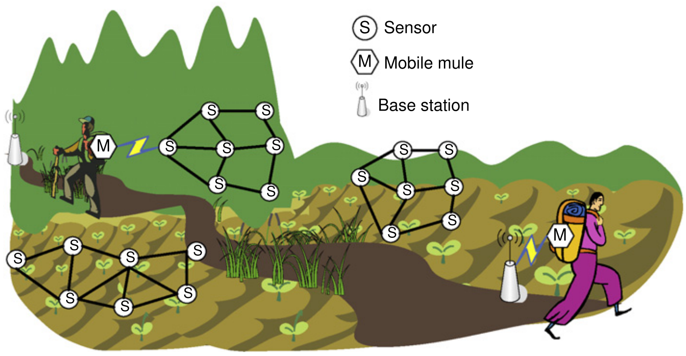


Fig. 1. A scenario of data collection for disconnected WSNs by mobile mules through opportunistic communications.

world. It is thus necessary to dispatch some *mobile mules* [10] to visit them from time to time and carry their sensory data back. WSNs may become isolated due to many reasons, such as physical constraints, cost considerations, and node failure owing to destructive events. In particular, our work is motivated by some recent work [11–14]. In [11], wireless sensors are used for in situ tracking of debris flows in wild mountain areas which are hard to reach by vehicles or human. Collecting data from such isolated WSNs thus may rely on rangers or hikers (when they reach those areas) in an opportunistic way to relay the sensing data back, as shown in Fig. 1. In the YushanNet Project [14] designed for YuShan National Park, Taiwan, hikers are used as opportunistic vehicles to relay sensing data back to outside world [13]. In [12], data collection, storage, and retrieval strategies for underwater WSNs are studied to monitor undersea oil fields. In all of these applications, it is hard to collect real-time information from those WSNs. So mobile mules, which could be animals, hikers, ships, or vehicles, are adopted in an opportunistic way to help collect and carry sensing data back.

In this paper, we consider a network scenario with three components: (i) some static but disconnected WSNs, (ii) some mobile mules with uncontrollable mobility, and (iii) some static base stations (BSs) accessible by mules. By ‘‘uncontrollable mobility’’, we mean that mules have their own designated routes or destinations that are not under control of our system (such as hikers, taxis, buses, and animals). Therefore, communications must rely on opportunistic contact between these entities. There are four types of communications in our system: (i) inter-node communications within a WSN, (ii) opportunistic WSN-to-mule communications, (iii) opportunistic mule-to-mule communications, and (iv) opportunistic mule-to-BS communications. Opportunistic communications happen when two entities have direct contact. We assume that each BS has connectivity to the external world, so our goal is to deliver sensing data to BSs.

The aforementioned networking scenarios raise several challenges to *storage management* and *opportunistic communications*. In disconnected WSNs, since sensors' memory spaces are limited and data collection from isolated WSNs to mules and then to BSs relies on opportunistic communications in the sense that contact between these entities is occasional, storing and collecting higher-priority data (e.g., the freshest and the most urgent data) is necessary. Two critical issues namely, the *data storage management* in each isolated WSN and *opportunistic data collection* between these entities, need to be addressed. For storage management, since an isolated WSN may not be visited by mules frequently, how to buffer more important data in the limited storage of a WSN is an important issue. The memory spaces of an isolated WSN can be regarded as a distributed storage system. The node that is more frequently visited by mules will be identified as the sink of a WSN. We then address three storage management issues: (i) how to buffer sensory data to reduce data loss due to a shortage of storage, (ii) if dropping of data is inevitable, how to avoid more important data from being dropped, and (iii) how to manage the data nearby the sink to facilitate the downloading jobs of mules. Note that (iii) is to facilitate opportunistic WSN-to-mule communications because the WSN-mule contacting time is unpredictable. For opportunistic communications, we assume that mobile mules have unlimited storage spaces, but the frequency and intervals of WSN-to-mule, mule-to-mule, and mule-to-BS contact are not under the control of our system. Therefore, the data exchange policy needs to be addressed when two entities have contact. Since a piece of sensing data needs to be carried from a sensor node to a sink, from a sink to a mule, from a mule to perhaps multiple mules, and from a mule to a BS, we regard its successful delivery to a BS as a reward and our goal is to accumulate more rewards at shorter time.

To respond to these challenges, we propose a *Distributed Storage Management* (*DSM*) strategy for data buffering in an isolated WSN and an *Opportunistic Data Exchange* (*ODE*) strategy for the occasional contact between two mules. DSM is designed based on a novel shuffling mechanism similar to heap sort [15] to keep data with higher priorities closer to the sink. However, unlike heap sort, which is based on a tree structure, DSM uses a mesh-like structure to facilitate data exchanges.[1] On the other hand, ODE

[1] Note that heap sort must be conducted in a complete binary tree. Insertion begins at a leaf and moves up toward the root, while deletion begins by removing the root element, moving the rightmost leaf element to the root, and then adjusting the heap. These operations are basically centralized operations and cannot be applied directly to a real distributed WSN environment.

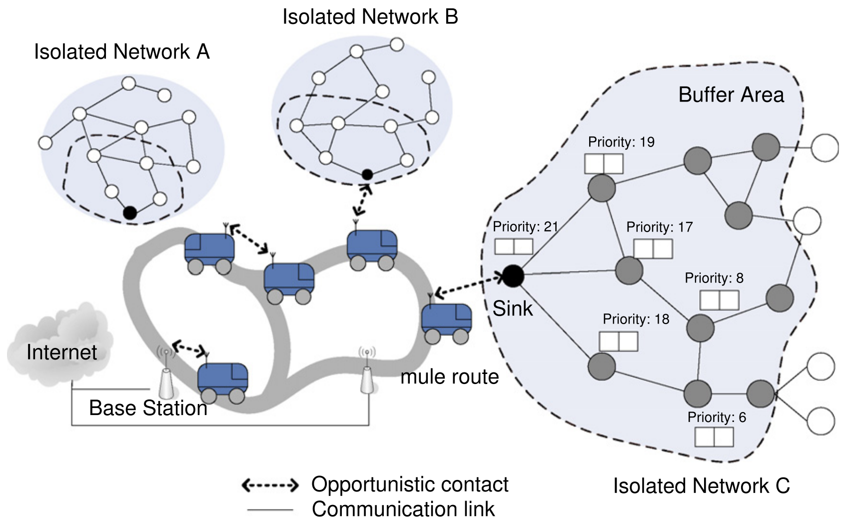


Fig. 2. Network model and communication architecture.

derives a probabilistic model to guide the mule-to-mule data exchanges so as to maximize the expected reward of delivering sensing data to BSs.

In the literature, using mules for data collection is addressed in [10,16,17]. Ref. [10] investigates the use of mules to connect sparse sensor networks with a three-tier architecture. Ref. [16] analyzes the upper bound of the optimal data transfer with mules. Ref. [17] shows that using mules with predictable mobility can significantly reduce communication power in WSNs. Ferry-assisted routing in a highly disconnected ad hoc network is discussed in [18–20]. A comprehensive survey of mobile sensor networks is in [21]. On the other hand, opportunistic communications are addressed in [22–26]. Opportunistic data forwarding schemes in a delay-tolerant/disruption-tolerant network are proposed in [22–24]. In an opportunistic mobile sensor network, reference [25,26] uses data redundancy to address the data gathering issue. Ref. [25] builds a connectionless tracking system to search and rescue a lost hiker, where sensors worn on hikers must exchange their *witness information* (including encounter time and most recently location) once they can communicate with each other. Then, each sensor on a hiker reports its witness to any access point deployed in the mountain area. In a loosely connected mobile sensor network, Ref. [26] proposes two data delivery schemes to decide when and where sensors should transmit their sensing data based on delivery probability and fault tolerance (i.e. data redundancy).

To the best of our knowledge, this is the first work addressing distributed storage strategy for isolated WSNs with opportunistic communications using mobile mules. The contributions of this work are as follows. First, we try to improve the data quality of collected packets in extreme sparse and resource-limited distributed WSNs. Second, we propose a novel concept to model the distributed storage of a WSN that could facilitate virtualizing the data management among sensor nodes' buffers. Third, we propose an opportunistic data exchange model which could improve the data quality of collected packets. Finally, we conduct extensive simulations to investigate the merit of our proposed strategies. The simulation results indicate that the level of data importance collected by our DSM is very close to a global optimization and our ODE could facilitate delivery of important data to BSs through mules. We also implement these strategies in a real sensor platform to verify the feasibility. Our prototyping results demonstrate that a lightweight implementation of these strategies is possible.

The rest of the paper is organized as follows. Section 2 presents our system model. Our DSM and ODE strategies are presented in Section 3 and Section 4, respectively. Section 5 contains our simulation results. Our implementation results are presented in Section 6. Section 7 concludes this paper.

2. System model

We consider a heterogeneous WSN consisting of three components: (i) some static but disconnected WSNs, (ii) some mobile mules with uncontrollable mobility, and (iii) some static base stations (BSs) accessible by mules. Each isolated WSN is composed of some *static sensor nodes*, or simply *nodes*, which can continuously monitor the environment and periodically generate *reporting packets*, or simply *packets*. Sensor nodes are homogeneous and each has the same number of storage spaces of S_{sn} (in unit of packet). Multi-hop routing is supported in each WSN. However, since these WSNs are deployed in remote fields and are isolated from the outside world, they rely on mobile mules to visit them and carry their sensory data out. We assume that each isolated WSN has a designated *sink* node that will be visited by mules occasionally. However, we assume that the movements of these mules are *uncontrollable*, i.e., they have their own routes or destinations which are not under control of our system (such as hikers, taxis, buses, and animals). A mule may stop by a WSN at any time and leave at any time. Therefore, communications only happen by opportunity. During this period, the sink should relay more important sensing data to the mule first.

In our system, there are four types of communications: (i) inter-node communications within a WSN, (ii) opportunistic WSN-to-mule communications, (iii) opportunistic mule-to-mule communications, and (iv) opportunistic mule-to-BS communications. Opportunistic communications happens when two entities have communication contact. We assume that all of these entities (sensor nodes, mules, and BSs) have the same communication interface. Any two entities can communicate with each other if they are within each other's transmission distance. Therefore, a piece of sensing data needs to be delivered from its originating sensor node to its sink, from the sink to a mule, from a mule to perhaps multiple mules, and finally from a mule to a BS. Fig. 2 shows our network model and communication architecture.

Since the communication scenarios in our system involve multiple relaying activities, we need to design a data storage strategy for WSNs and an opportunistic data collection strategy among mules so that more important data could be delivered to BSs. In our system, we assume that packets generated by sensor nodes are prioritized according to their importance. For example, a fire report in a forest area is more important than a smoke report, a smoke report is more important than a high temperature report, a high temperature report is more important than a regular temperature report, etc. Reports in different locations may be prioritized too. For example, a status report of a bridge is more important than one of a regular road. Also, an aging process may be applied to the priority of a report. For example, a temperature report of 100 F an hour ago may be less important than a temperature report of 95 F a minute ago. We will use a function $f(p)$ to denote the priority of a packet p. How to design function $f()$ is application-dependent and is beyond the scope of this work.

In the following sections, we will propose two strategies, called *Distributed Storage Management* (*DSM*) and *Opportunistic Data Exchange* (*ODE*) to meet the above challenges.

3. The DSM strategy

Below, we consider only one isolated WSN and focus on its storage management problem. All WSNs will follow the same strategy to store their sensing data.

Given a WSN, we assume that there is a predefined region nearby its sink called the *Buffer Area* (*BA*).[2] The set of nodes in BA will act as a distributed storage system to store sensing data for the WSN when no mule is visiting it. When a mule arrives, these nodes will forward their data to the sink following some rules (see **E1–E3.2** below). In Fig. 2, the BA of WSN C contains nodes within three hops from the sink. (In an extreme case, one may designate all of nodes in WSN as the BA.) We also assume that each node u knows its distance $D(u)$ to the sink and its neighbor set $N(u)$. (To obtain $D(u)$, a simple broadcast from the sink node can achieve this goal. Also, $N(u)$ can be found by exchanging hello messages among neighbors.)

All static sensor nodes will try to forward their packets toward nodes in BA at any time. Regarding the storage spaces in BA as a distributed storage system, our goal is to design a distributed protocol to achieve three objectives.

- **G1.** Dropping of packets in BA should be minimized.
- **G2.** If dropping of packets is unavoidable, those with lower priorities should be dropped first.
- **G3.** To facilitate mobile mules to collect data, packets with higher priorities should be stored closer to the sink.

Definition 1. Given a WSN represented by a graph $G = (V, E)$, its buffer area $BA \subseteq V$, and a priority function $f()$, the Distributed Storage Management (DSM) problem is to develop a packet exchange protocol to maintain packets being generated by the WSN within BA such that properties G1–G3 are met and $\Omega(BA) = \sum_{v \in BA, p \rightarrow v} f(p)$ is maximized, where $p \rightarrow v$ means that a packet p is stored at the storage of v.

The objective function $\Omega(BA)$ reflects our goal of accumulating packets with higher priorities inside the BA region. Our DSM strategy is a distributed solution based on a shuffling mechanism. Nodes not in BA will forward their packets to BA, while nodes in BA will observe neighbors' states and exchange packets with each other, if necessary.

Without loss of generality, we assume that each node u has only one buffer space, i.e., $S_{sn} = 1$. (Our scheme can be easily extended to $S_{sn} > 1$.) So the (only) packet in u is written as $P(u)$ and its priority is $f(P(u))$ (if u has no packet, $f(P(u)) = -1$). DSM tries to maintain the following properties for each node $u \in BA$.

- **P1.** For each node $v \in N(u)$ such that $D(v) > D(u)$, $f(P(v)) \leqslant f(P(u))$.
- **P2.** For each node $v \in N(u)$ such that $D(v) < D(u)$, $f(P(v)) \geqslant f(P(u))$.
- **P3.** For each node $v \in N(u)$ such that $D(v) = D(u)$, $max\{f(P(w)) | w \in N(u), D(w) > D(u)\} \leqslant f(P(v)) \leqslant min\{f(P(w)) | w \in N(u), D(w) < D(u)\}$.

P1 (resp., **P2**) implies that nodes that are farther from (resp., closer to) the sink than u should have lower-priority (resp., higher-priority) packets than u. **P3** enforces that nodes that have the same distance to the sink as u should have the same properties as u. When a node has the above properties, we say that it is *in-order*. In Fig. 3a, every node is in-order except node m and j.

For each node u, we let $maxPost(u)$ be the packet with the highest priority of all neighbors v of u such that $D(v) > D(u)$, $minPre(u)$ be the packet with the lowest priority of all neighbors v of u such that $D(v) < D(u)$, $maxEqual(u)$ be the packet with the highest priority of all neighbors v of u such that $D(v) = D(u)$, and $minEqual(u)$ be the packet with the lowest priority of all neighbors v of u such that $D(v) = D(u)$. Here, to facilitate the determination of

[2] The size of BA depends on the application context. In a long-term monitoring system [27], assuming that each sensor node has data arrival rate of λ, the interval between two consecutive visits is roughly T, and N is the number of nodes in an isolated WSN, the size of BA may be set to $\min\{\frac{\lambda TbN}{S}, N\}$ if we want to store all fresh data between two visits, where b is the size of a packet and S is the total memory space of a node.

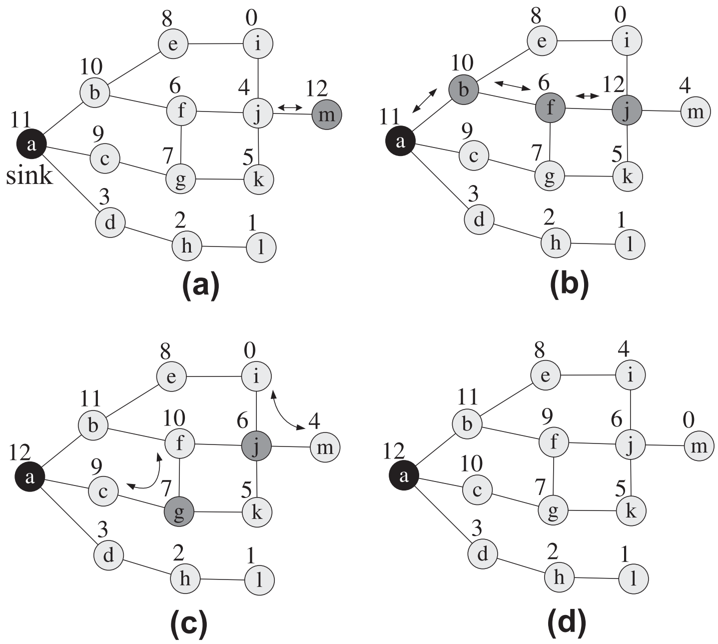


Fig. 3. An example of DSM packet exchanges.

$maxPost(u)$, $minPre(u)$, $maxEqual(u)$, and $minEqual(u)$, each node will announce the priority of its packet when a new packet is generated. It implies that our protocol is light-weight and can be implemented easily via local information exchanges. Based on the above properties, we design our packet exchange rules for node $u \in BA$ as follows:

- **E1.** When $f(maxPost(u)) > f(P(u))$, node u tries to exchange packet with $maxPost(u)$.
- **E2.** When $f(P(u)) > f(minPre(u))$, node u tries to exchange packet with $minPre(u)$.
- **E3.1.** When $f(maxEqual(u)) > f(minPre(u))$, these two packets are exchanged.
- **E3.2.** When $f(maxPost(u)) > f(minEqual(u))$, these two packets are exchanged.

The above rules are event-triggered ones. They are triggered when a node changes its packet (including exchanging its packet with others' or generating a new packet) or when its neighbors change their packets. When multiple events are triggered, a node should prioritize rules **E1, E2, E3.1**, and **E3.2** in that order because we prefer nodes exchanging with those at different distance first. For node u to exchange packet with node v, it can send a *Request_To_Exchange (RTE)* to node v. Node v, on agreeing, replies a *Clear_To_Exchange (CTE)*. Then the exchange can be conducted. These operations should be atomic. Note that u and v are not allowed to exchange packets with other nodes during the exchange of RTE and CTE to ensure atomic transactions.

For a node $u \notin BA$, when it has a packet, it will try to send it to any neighbor v with $D(v) < D(u)$. When a node $w \in BA$ receives the packet, it will accept it if $f(P(w)) = -1$, drop it if $f(P(w)) \geqslant f(P(u))$, and replace $P(w)$ by $P(u)$ if $f(P(u)) > f(P(w))$.

We provide an example in Fig. 3. Node a is the sink and there is a new packet with priority 12 arriving at node m in Fig. 3a. Node m will realize that it violates **P2** and will exchange with node j by **E2** as shown in Fig. 3b. The same situation will happen to nodes j, f, and b, resulting in the scenario in Fig. 3c. Now j finds that it violates **P3** because $f(P(i))$ is not between 10 and 4. So j will notify i and m to exchange their packets by **E3.2**. Similarly, g will find that it violates **P3** after receiving f's broadcast and notify c and f to exchange their packets by **E3.1**. The final result is in Fig. 3d, where every node is in-order. Note that DSM does not guarantee an optimal arrangement of packets since it is a distributed protocol and relies only on neighbors' information.

3.1. Proof of correctness

Below, we prove that DSM will eventually stop in an in-order status. We say that a packet is *stable* if it is stored in a certain node and will not be exchanged with other nodes' packets, until a mule arrives or new packets with higher priority are being generated. We first show that each packet will become stable in finite time, which means that DSM will eventually stop. Then we show that each node is in-order when DSM stops.

Theorem 1. *Given any arrangement of packets in BA of a WSN, if no mule arrives and no packets are being generated, the packet exchange rules* **E1**, **E2**, **E3.1** *and* **E3.2** *will eventually stop in finite time.*

Proof. It is obvious that the packet with the highest priority eventually migrates to the sink and becomes stable once it reaches the sink. Once the packet with the highest priority becomes stable, the packet with the second-highest priority can become stable once it reaches a neighboring node of the sink. Similarly, each packet can become stable if all packets with higher priorities than it have become stable and it reaches a place as close to the sink as possible. Note that it is not necessary that packets become stable in the order of their priorities. But once higher prioritized packets become stable, a packet can become stable without doubt. Since the region of BA is limited, the packet exchange activities will stop in finite steps. □

Theorem 2. *After all nodes in BA stop exchanging packets, they are in-order.*

Proof. We prove this theorem by contradiction. If node u is not in-order, then there are only three possible cases:

Case 1: Node u violates **P1**. That is there is a neighbor $v \in N(u)$ such that $D(v) > D(u)$ and $f(P(v)) > f(P(u))$. Since $f(maxPost(u)) \geqslant f(P(v)) > f(P(u))$. It will not stop exchanging packets according to **E1**.

Case 2: Node u violates **P2** but it follows **P1**. That is there is a neighbor $v \in N(u)$ such that $D(v) < D(u)$ and $f(P(v)) < f(P(u))$. Since $f(minPre(u)) \leqslant f(P(v)) < f(P(u))$, it will not stop exchanging packets according to **E2**.

Case 3: Node u violates **P3** but it follows **P1** and **P2**. That is there is a node $v \in N(u)$ such that $D(v) = D(u)$ and the value of $f(P(v))$ is not between $f(maxPost(u))$ and $f(minPre(u))$. Since node u follows **P1** and **P2**, we have $f(minPre(u)) \geqslant f(maxPost(u))$. So the value of $f(P(v))$ is either larger than $f(minPre(u))$ or smaller than $f(maxPost(u))$.

- $f(P(v)) > f(minPre(u))$: Since $f(maxEqual(u)) \geqslant f(P(v)) > f(minPre(u))$, it will not stop exchanging packets according to **E3.1**.
- $f(P(v)) < f(maxPost(u))$: Since $f(minEqual(u)) \leqslant f(P(v)) < f(maxPost(u))$, it will not stop exchanging packets according to **E3.2**.

Cases 1–3 all contradict to our assumption that nodes have stopped exchanging packets, so this theorem is proved. □

To summarize, DSM utilizes the rich mesh links in a WSN to exchange packets. Higher-priority packets have more chances to stay closer to the sink by rules **E3.1** and **E3.2**. One question is: given a stable network, how many packet exchanges may be incurred when a new packet is generated. We will investigate this issue via simulations.

3.2. Two extensions to the DSM strategy

Below, we discuss two extensions to the above DSM strategy. First, we enlarge the value of S_{sn}. Second, we discuss the possibility of adding some transmission buffers to sensor nodes.

To allow $S_{sn} > 1$, we define $maxMine(u)$ (resp., $minMine(u)$) to be the packet of u with the highest (resp., the lowest) priority. Since a node may have multiple packets, the exchange rules **E1** and **E2** for node u are modified as follows:

E1′. When $f(maxPost(u)) > f(minMine(u))$, node u tries to exchange its packet $minMine(u)$ with packet $maxPost(u)$.

E2′. When $f(maxMine(u)) > f(minPre(u))$, node u tries to exchange its packet $maxMine(u)$ with packet $minPre(u)$.

Rules **E3.1** and **E3.2** do not need to be changed to allow $S_{sn} > 1$. The definition of "in-order" can be directly applied to $S_{sn} > 1$.

The second extension is to add a few transmission buffers to each node to handle packet overflow. Our DSM strategy may enter a dilemma when a node already holds S_{sn} packets and generates a new packet by its own sensors; either this packet or one of its existing S_{sn} packets needs to be dropped because there is no extra buffer space. Transmission buffers are designed for this purpose. A packet waiting to be transmitted should be put in a transmission buffer. When a node $u \in BA$ with S_{sn} packets at hand generates a new packet, it will keep S_{sn} packets with higher priorities in its storage spaces and move the lowest-priority one to its transmission buffer. The lowest-priority packet will be forwarded to the neighbor $v \in N(u)$ which has $D(v) > D(u)$ and $f(minMine(v)) < f(minMine(w))$, $v, w \in N(u)$ so as to replace v's packet with the lowest priority (i.e., $f(minMine(v))$) in case v also holds S_{sn} packets. Note that each node only needs to announce the highest and the lowest priorities of its packets to its neighbors in the above two extensions. Thus, all of previous properties (both Theorems 1 and 2) still hold when $S_{sn} > 1$.

4. The ODE strategy

As mentioned earlier, there are three types of opportunistic communications in our system: WSN-to-mule, mule-to-mule, and mule-to-BS. We explain the data collection strategy for each type of opportunistic communications as follows.

First, for WSN-to-mule communications, since packets in BA are already in-order by our DSM, a simple best-effort uploading strategy will serve our data-collection goal. Specifically, when a mule arrives at the sink of a WSN, the sink will try to transmit as many packets in BA to the mule as possible until it loses the contact with the mule. By broadcasting an UPLOAD message, the sink will trigger data transmission from downstream nodes toward itself in a greedy way. After the sink makes sure the reception of a packet by the mule, it can drop the packet so as to make a space for subsequent packets. Once the sink loses the contact with the mule, it will broadcast a FINISH_UPLOADING message to trigger our DSM in the WSN.

Next, we need to model the mule-to-mule communications. Since each mule may has different probabilities to contact with BSs in the future, we design a *utility-based* packet exchange strategy based on the contact probability to guide data exchanges between mules so as to maximize the reward of packets arriving at BSs. Specifically, when mule u meets mule v, a packet will be copied from u to v if the benefit of copying the packet from u to v is larger than the benefit of copying another packet from v to u. To achieve the goal, we need to design a benefit model. First of all, we need to model the distribution that a mule will have contact with a BS in the future. Suppose that two mules u and v meet at time t. Let $h_u(t+\Delta t)$ and $h_v(t+\Delta t)$ be the discrete probability distributions of u and v, respectively, that they will have contact with the next BS at time $t+\Delta t$, where $\Delta t > 0$ is an integer. Our derivation allows $h_u(t+\Delta t)$ and $h_v(t+\Delta t)$ to be any general distributions. However, in reality, this should depend on the time when u and v met a BS previously before t.

Now, suppose that u has a packet p at hand. Then at time $t+1$, the probability that p can be delivered from u to a BS is $h_u(t+1)$. If at time t mule u decides to transmit a copy of p to mule v, then v can also help deliver p to a BS. The probability that this will happen at time $t+1$ is $h_v(t+1)$. Now since both u and v have a copy of p, the cumulative probability that p can be delivered to a BS at time $t+1$ becomes

$$H(t+1) = 1 - (1 - h_u(t+1)) \cdot (1 - h_v(t+1)).$$

For any Δt, if at time t mule u transmits a copy of p to mule v, the joint cumulative probability that p will be delivered to a BS by time $t+\Delta t$ is

$$H(t+\Delta t) = 1 - \left(1 - \sum_{t'=1..\Delta t} h_u(t+t')\right) \cdot \left(1 - \sum_{t'=1..\Delta t} h_v(t+t')\right),$$

where $(1 - \sum_{t'=1\cdots\Delta t} h_u(t+t')) \cdot (1 - \sum_{t'=1\cdots\Delta t} h_v(t+t'))$ evaluates the probability that p is not delivered to a BS by both mule u and mule v before $t+\Delta t$. Therefore, the probability that the first piece of p will be delivered to a BS by u or v at time $t+\Delta t$ is

$$H'(t+\Delta t) = H(t+\Delta t) - H(t+\Delta t - 1).$$

Next, we derive our utility model. Recall the priority of a packet, denoted by $f()$. Given any packet p at time t, if p will be delivered to a BS at time $t+\Delta t$, we define a utility function of p as $\Theta(f(p),\ a(t+\Delta t), c(p)) = f(p) \times \alpha^{a(t+\Delta t)} \times \beta^{c(p)}$, which is a decay function of priority over time and the estimated number of copies of packet p, where $f(p)$ is the priority of p, $a(t+\Delta t)$ is the age of packet p at time $t+\Delta t$, and $c(p)$ is the estimated number of copies of packet p. Here, $0 < \alpha < 1$ is a decay coefficient for packet aging and $0 < \beta < 1$ is a decay coefficient as the number of copies of p increases. Thus, $\Theta(f(p),\ a(t+\Delta t), c(p))$ will return a positive value to reflect the level of satisfaction when p is delivered to a BS at time $t+\Delta t$. Note that function $\Theta()$ can avoid collecting too old packets and too many duplicates of a packet. Therefore, if at time t mule u decides to transmit a copy of p to mule v, the expected utility is

$$E(u \to v, p, t) = \sum_{\Delta t \geqslant 1} H'(t+\Delta t) \cdot \Theta(f(p), a(t+\Delta t), c(p)+1).$$

On the contrary, if at time t mule u decides not to do so, the expected utility is

$$E(u, p, t) = \sum_{\Delta t \geqslant 1} h_u(t+\Delta t) \cdot \Theta(f(p), a(t+\Delta t), c(p)).$$

The benefit of copying p from u to v is thus

$$E(u \to v, p, t) - E(u, p, t).$$

Based on the benefit derivation, ODE works as follows when two mules u and v have contact.

1. Each mule will consider whether copy packets from itself to another by sorting its packets based on the current utility of packets.
2. Mule u considers the packet, say, p at hand which has the highest utility and which has not been considered yet. Also, v considers the packet, say, q at hand which has the highest utility and which has not been considered yet. Then both u and v compute the benefits $E(u \to v, p, t) - E(u, p, t)$ and $E(v \to u, q, t) - E(v, q, t)$, respectively.
3. If copying p makes a higher benefit, u copies p to v; otherwise, v copies q to u.
4. The packet that is copied is marked as "considered". If u and v are still within each other's communication range, go to step 2. Otherwise, stop.

Note that we assume a CSMA channel, so concurrently copying both p and q in step 3 is not allowed. Also note that since the contact duration between mules is unpredictable, a best-effort copy policy is applied here.

Finally, for mule-to-BS communications, when a mule has contact with a BS, a best-effort uploading strategy is applied by the mule to transmit packets of higher utility first.

5. Simulation results

In this section, we will explain the simulation setup first and then conduct extensive simulations to demonstrate the merit of DSM and ODE.

5.1. Simulation setup

We built our simulator in Java programs. Unless otherwise indicated, the simulation scenarios and the default values of parameters in our simulations are explained as follows.

To investigate the performance of DSM, we simulate an isolated WSN which contains 400 sensor nodes randomly deployed in a $200 \times 200\ \text{m}^2$ field, where the sink is randomly selected. Each sensor node has a communication range of 25 m.[3] Each sensor node generates a packet with a random priority uniformly distributed between 0 and 1000 at a packet arrival rate of 1/50 Hz.[4] The BA is defined as those nodes within 10 hops from the sink. A mule is

[3] The typical ZigBee communication range is generally between 10 $\sim$ 100 m [28].

[4] In a weather monitoring application, temperature is measured roughly every 10 s to several minutes, where the measurement interval is restricted by the memory size of monitoring units [29].

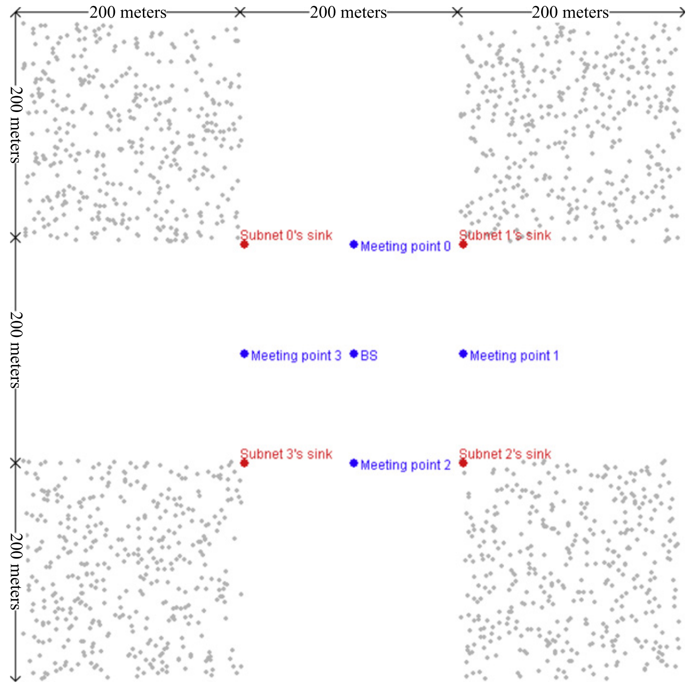


Fig. 4. The deployment for ODE simulations with four isolated WNSs and four meeting points of mules.

deployed to collect data from the WSN with a fixed vising period of 100 communication slots each time with a fixed contact duration of 30 communication slots (here we assume that a communication slot is a short period of time such that one packet transmission can be completed). We take the average of 50 test runs to present our simulation results. First, we run DSM using both a mesh structure and a tree structure to understand how the topology has an effect on performance. Then, we compare DSM against two different strategies, *Greedy Forward* (*GF*) and *optimal data storage scheme* (*OPT*). In GF, a node always tries to send its packets to any node closer to the sink until the latter has no storage space. OPT represents the ideal situation where the top-priority packets are always retained in BA.

To study the performance of ODE, we deploy four isolated WSNs and four meeting points of mules in a $600 \times 600\,m^2$ field, as shown in Fig. 4. A BS is deployed at the center of the field. The BA in each WSN includes those sensor nodes within 10 hops from its sink, and each WSN will perform DSM to conduct data collection. We deploy 2 mules in the field, each being placed at the BS initially. Each mule will visit any one of the four WSNs every 125 s (i.e., vising period is 125 s) to collect data from the WSN. To simulate encounters between mules, each mule has a probability of 0.5 to move to any one of the four meeting points at any time. If two mules meet at a meeting point, they will have the contact duration of 30 communication slots to perform ODE. To simulate contact between mules and the BS, each mule will meet the BS every 40 s (i.e., the meeting period is 40 s), where the meeting probability during each period is a normal distribution with the variance 1. Every time when a mule meets the BS, the contact duration is 30 communication slots. The total simulation time is 500 s. Each packet will increase its age by one every 10 s. We compare ODE against a *Greedy Copy Scheme (Greedy)* to study the performance of ODE. In Greedy, when packet p hold by mule u and packet q hold by mule v are considered at the contact moment, p will be copied from u to v if p has the higher priority than q.

5.2. Performance of DSM

We conduct extensive simulations to study the merit of our DSM strategy.

First, we present the priority distribution in a WSN after applying DSM, as shown in Fig. 5, where the sink is at (0,0) and the bar standing next to the graph indicates different levels of packet priorities.

In the second experiment, we perform DSM using both a mesh structure and a tree structure to study how the

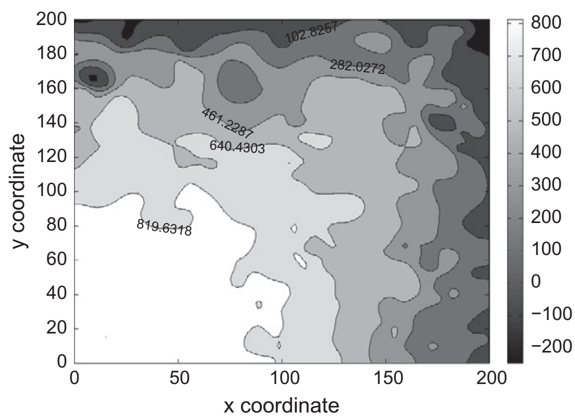


Fig. 5. A snapshot of priority distribution after applying our DSM strategy.

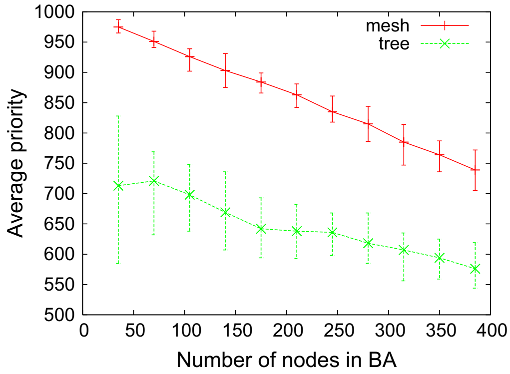


Fig. 6. Comparison of applying DSM on a mesh structure and a tree structure.

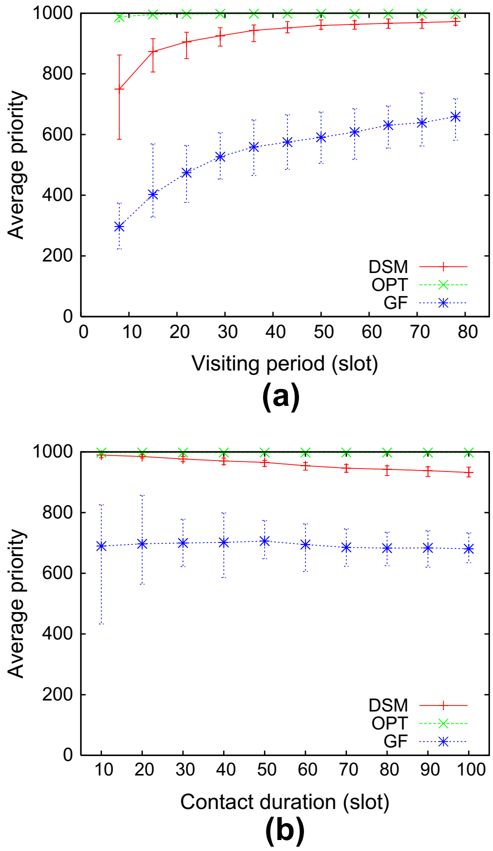


Fig. 7. Comparison of the average priorities of the collected packets by varying the mule's (a) visiting period (with the fixed contact duration of 30 communication slots) and (b) contact duration (with a fixed visiting period of 100 communication slots).

topology has an impact on data collection. To construct a tree structure, we simply form a shortest-path spanning tree rooted at the sink from the given mesh graph. In such a tree, each node is only allowed to exchange packets with its parent or children. Fig. 6 shows the average priority of packets at nodes in BA by varying the number of nodes in BA. Clearly, although the hop distance from each node to the sink is same in the shortest-path spanning tree of

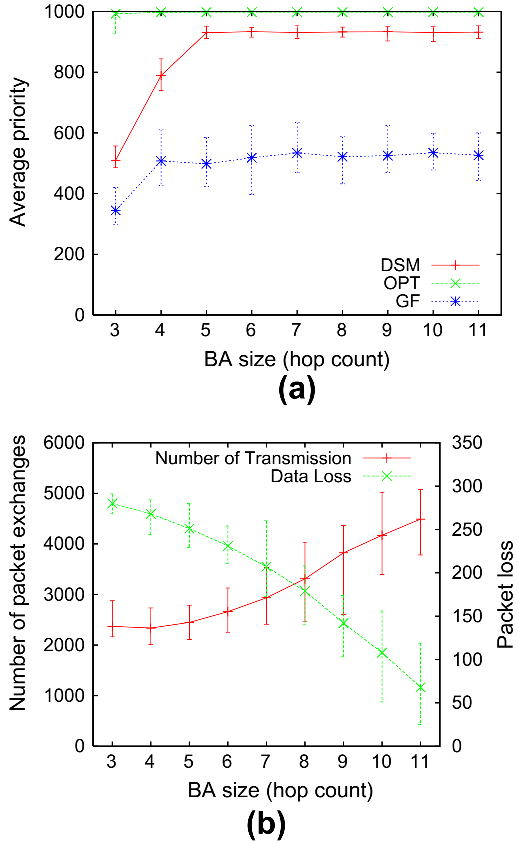


Fig. 8. The effects of BA size on (a) average priority of the collected packets and (b) transmission overhead and number of dropped packets.

the mesh and the mesh itself, using a mesh structure has potential to collect higher-priority packets than using a tree structure. This is because the mesh structure allows much more directions of data exchanges which will keep more higher-priority packets in BA.

In the third experiment, we try to compare the average priorities of the packets collected by the mule using different strategies by varying the mule's visiting period and the contact duration. Fig. 7a shows the results when we vary the visiting period. Generally, a longer visiting period means that more important packets may be generated/collected during two consecutive visits. So we can see that curves are going up. DSM significantly outperforms GF and is quite close to OPT. Fig. 7b shows the results when we vary the contact duration. Generally, the longer contact duration means that less important packets also have a chance to be collected, so we can see that curves are slightly going down. Again, DSM is still much better than GF and is quite close to OPT.

In the fourth experiment, we study how the size of BA has an effect on performance. We vary the size of BA but enforce that in each contact between the mule and the WSN, 1/3 of the packets in the network must be collected. Fig. 8 shows the effects of BA's size (in terms of hop counts to the sink). As it can be seen in Fig. 8a, with our DSM strategy, the top 1/3 area in the network are mostly occupied by high-priority packets as BA is defined as five hops or more

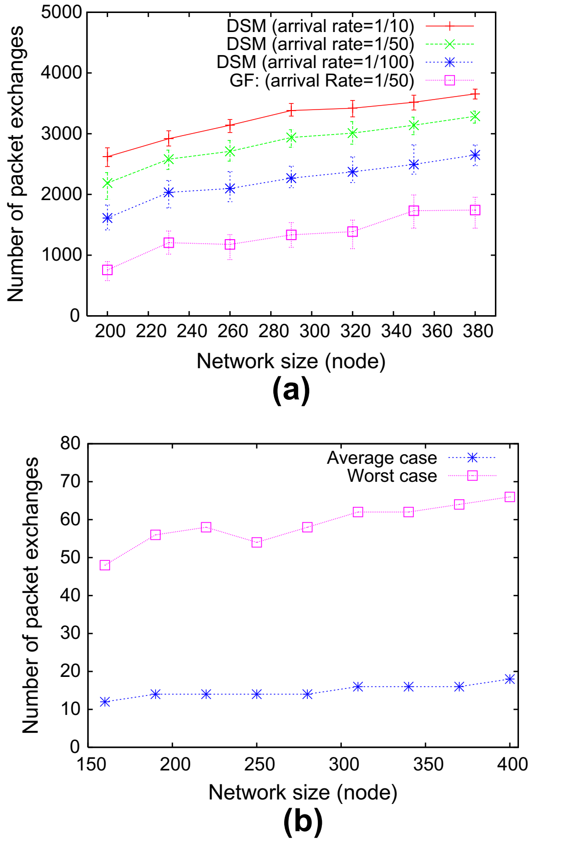


Fig. 9. Comparison of transmission overhead and packet exchanges under various network sizes: (a) overall transmission overhead and (b) number of packet exchanges when a new packet arrives at a stable BA region.

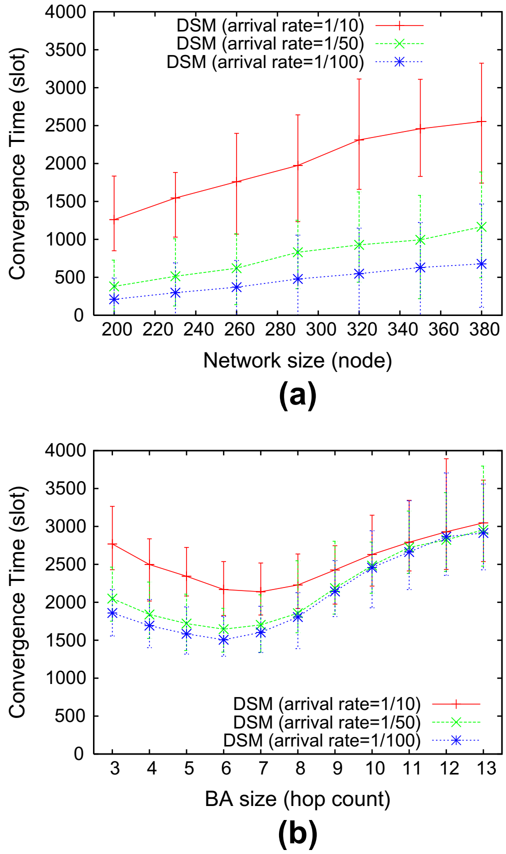


Fig. 10. Comparison of convergence time: (a) convergence time vs. network size. (b) convergence time vs. BA size.

from the sink and this is very close to OPT. Fig. 8b shows the transmission overhead (in terms of the number of packet exchanges) and the average number of dropped packets as we vary the BA's size. Interestingly, the transmission overhead slightly decreases first and then increases when the BA is getting larger. This is because packets with lower priorities have to travel longer to reach the BA when the BA is relatively smaller (3–4 hops). However, transmitting such low-priority packets makes a little sense because they are more likely to be dropped as they arrive at the BA and compete against those higher-priority packets. As the BA becomes larger (more than five hops), the cost of packet exchanges inside the BA is more dominant compared to the aforementioned factor. So, we can see that the transmission overhead is increasing again. On the other hand, the number of dropped packets decreases as BA becomes larger, because a larger size of BA can keep more packets and avoid dropping packets when they arrives BA.

In the fifth experiment, we study how the network density has an effect on performance by changing the number of sensor nodes in the field. Fig. 9 shows the results. First, we observe the impact on overall transmission overhead in Fig. 9a. At the same packet arrival rate of 1/50, DSM costs about 0.8–1.1 times more packet exchanges than GF to collect more important packets in BA. On the other hand, we also see that increasing the packet arrival rate does not increase the overall transmission overhead proportionally (the overhead at a rate of 1/10 is only slightly higher than the overhead at a rate of 1/50). This is because once the BA has collected sufficiently important packets, the competition cost within BA will drop rapidly. In Fig. 9b, we try to measure the number of packet exchanges incurred when a new packet (with a random priority) arrives at a stable BA region (i.e., packets in BA are already well in-order). We count the number of packet exchanges in the average and the worst cases. As it can be seen, a denser network will cause a higher cost of packet exchanges because sensor nodes have more neighbors to facilitate data exchanges for the new packet. In general, the overhead is not high (10–20 exchanges in average).

In the sixth experiment, we study the convergence time of DSM by considering different data arrival rates. The convergence time is defined as the minimal time that all of packets in BA are in-order. We vary the number of nodes and the size of BA to study the convergence time. Fig. 10 shows the simulation results. As it can be seen in Fig. 10a, a dense network will incur a longer convergence time because more number of packet exchanges are performed. We can also see that the convergence time is not proportional to the data arrival rate, because packet exchanges in the network at a higher data arrival rate will

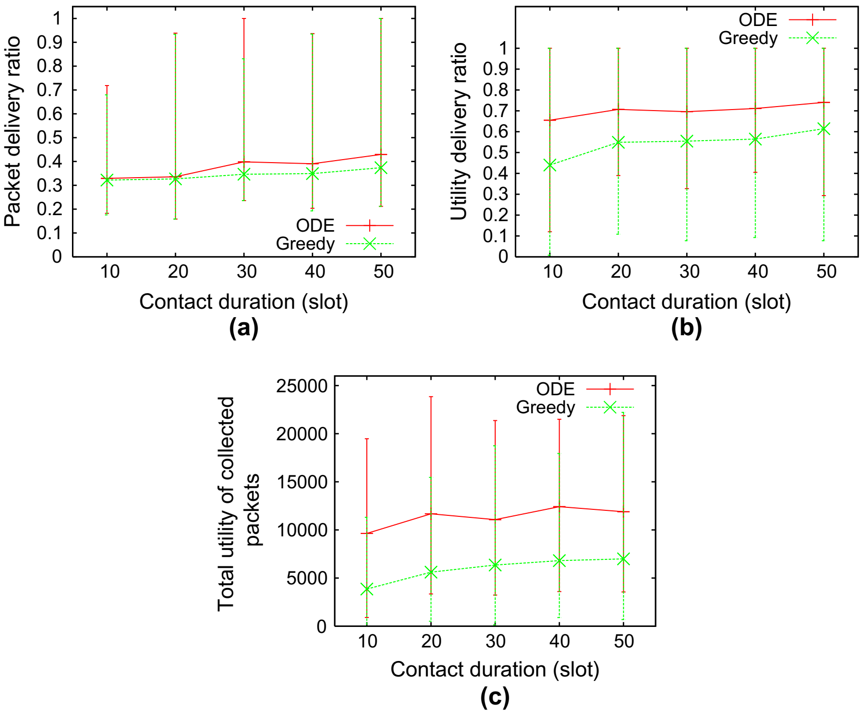


Fig. 11. Simulation results of ODE by varying the contact duration: (a) the packet delivery ratio, (b) the utility delivery ratio, and (c) the total utility of packets collected by the BS.

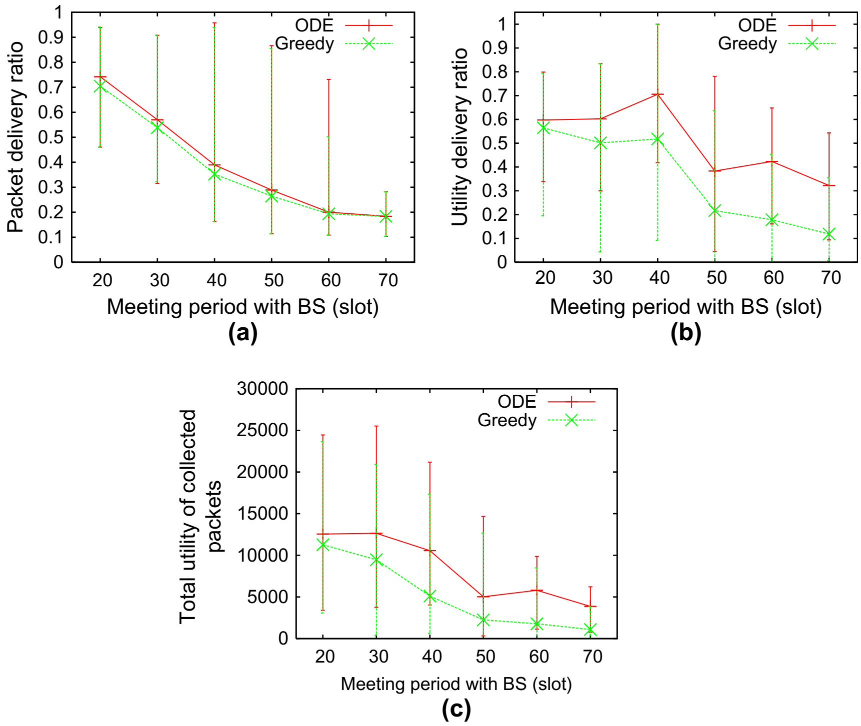


Fig. 12. Simulation results of ODE by varying the meeting period with the BS: (a) the packet delivery ratio, (b) the utility delivery ratio, and (c) the total utility of collected packets.

be triggered frequently. Fig. 10b shows that the convergence time slightly decreases first and then increases again as the BA's size increases. This is because in a relative small size of BA (from 3 to 7 hops) lower-priority packets have to move by more hops to compete against the higher-priority packets in BA. When the network has a relative large size of BA (from 8 to 13 hops), much more packet exchanges will incur in BA so the convergence time becomes longer. As it can be seen, the gaps of convergence time between different data arrival rates shrink as BA's size increases, because a larger BA provides more storage spaces to allow more concurrent packet exchanges.

5.3. Performance of ODE

Next, we will study the merit of ODE strategy by varying the mule-to-mule and mule-to-BS contact duration, the meeting period with the BS, and the aging period of packets. We consider three metrics to study the performance of ODE: (1) packet delivery ratio which is the ratio of the total number of packets collected by the BS to the total number of packets collected by mules, (2) utility delivery ratio which is the ratio of the total utility of packets collected by the BS to the total utility of packets collected by mules, and (3) the total utility of packets collected by the BS. Note that the latter two metrics are to study the performance from a perspective of information quality. The second one is a relative metric to study the improved degree of information quality using ODE, while the third one is an absolute metric to study the total amount of information quality improved by ODE. Both of the two metrics are necessary to know how well the information quality can be improved by ODE and the how ODE has an effect on the quality of information collected.

First, we vary the contact duration to conduct experiments. Fig. 11 shows the simulation results. As it can be seen in Fig. 11a, the packet delivery ratio by ODE is slightly better than by Greedy when the longer contact duration is considered. On the other hand, when the contact duration increases, the packet delivery ratio by ODE is improved, but the improvement in Greedy is not significant. This is because Greedy collects many duplicated higher-priority packets even if the longer contact duration is considered. Then, we study the utility delivery ratio, as shown in Fig. 11b, ODE outperforms Greedy because the packets collected by ODE are younger and have the fewer number of copies. In terms of total utility of packets collected by the BS, ODE also outperforms Greedy significantly, as shown in Fig. 11c.

Second, we study how the meeting period between mules and the BS has an effect on performance. We conduct the experiment by varying the meeting period. Note that the meeting probability between a mule and the BS within each period follows a normal distribution. Fig. 12 shows the simulation results. As shown in Fig. 12a, when a smaller meeting period is considered, the packet delivery ratio by ODE is slightly better than by Greedy. This is because duplicated packets are frequently collected by Greedy. As the meeting period increases, the gap between ODE and Greedy shrinks because mules have the fewer

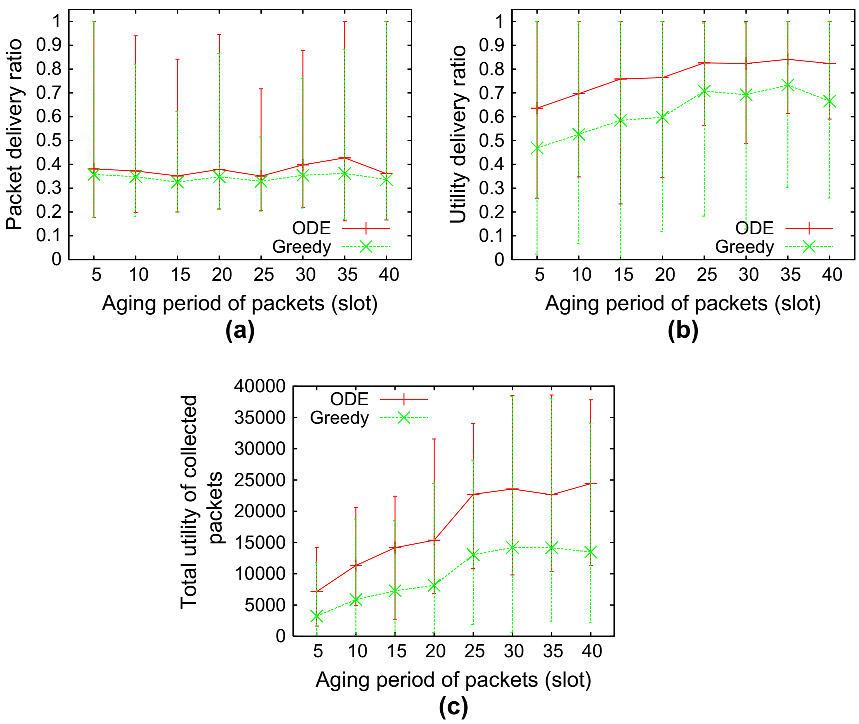


Fig. 13. Simulation results of ODE by varying the aging period of packets: (a) the packet delivery ratio, (b) the utility delivery ratio, and (c) the total utility of collected packets.

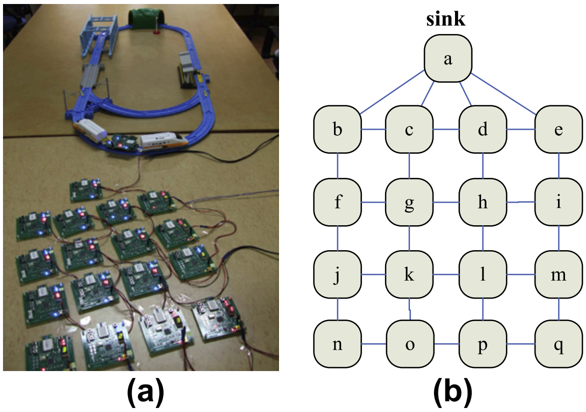


Fig. 14. Our implementation of DSM (a) a snapshot of our prototype and (b) a grid WSN.

opportunities to meet the BS. Then, we investigate how the meeting period has an effect on the utility delivery ratio, As shown in Fig. 12b, when the mules have fewer opportunities to meet the BS (i.e., with a larger meeting period), ODE outperforms Greedy significantly. This indicates that ODE can collect better packets (i.e., younger packets and with fewer number of copies) even if the frequency of meeting with the BS is very low. On the other hand, both by ODE and by Greedy, the utility delivery ratio decreases as the meeting period increases. This is because mules do not have so many opportunities to upload data to the BS when a larger meeting period is considered. Fig. 12c shows the results of the total utility of packets collected by the BS. Similarly, ODE outperforms Greedy especially for a larger meeting period.

Finally, we vary the aging period of packets to study how the aging speed of packets has an effect on performance. Fig. 13 shows the simulation results. As it can be seen in Fig. 13a, ODE is slightly better than Greedy in terms of packet delivery ratio because ODE avoids copying too many higher-priority packets. In Fig. 13b, the utility delivery ratio sightly increases as the aging period increases. Since packets' priorities decay in a slow speed when a larger aging period is considered, the packets collected by ODE and by Greedy have the better utility. Similarly, as shown in Fig. 13c, when a large aging period is considered, the gap between ODE and Greedy becomes larger due to slow decay of priorities.

6. Prototype for DSM

Finally, we have implemented DSM in a real sensor platform to verify the feasibility of our proposed strategies. A toy train is designed to repeatedly circle around a toy rail. The train serves as a mule, and we deploy a wireless node on it. A number of isolated grid WSNs are deployed around the rail. Whenever the train has contact with a sink, it will pull as many packets from the sink as possible. Fig. 14a shows a snapshot of our prototype. Our sensor hardware platform is a low-power, single-chip wireless microcontroller JN5139 [30] with a ZigBee-compliant wireless interface. The WSN in Fig. 14b is a 4 × 4 grid plus a sink. Each JN5139 runs our DSM strategy and can store one packet at a time. Each node will generate real sensing data using a light sensor, and each packet has a priority ranging from 0 to 9, where a higher priority means a higher light intensity. To view the priority of a packet, we put a 7-segment display on each sensor node. When the mule has contact with the sink, it initiates a COLLECT_DATA message to the sink. Then the sink broadcasts an UPLOAD message to its members as explained in Section 4.

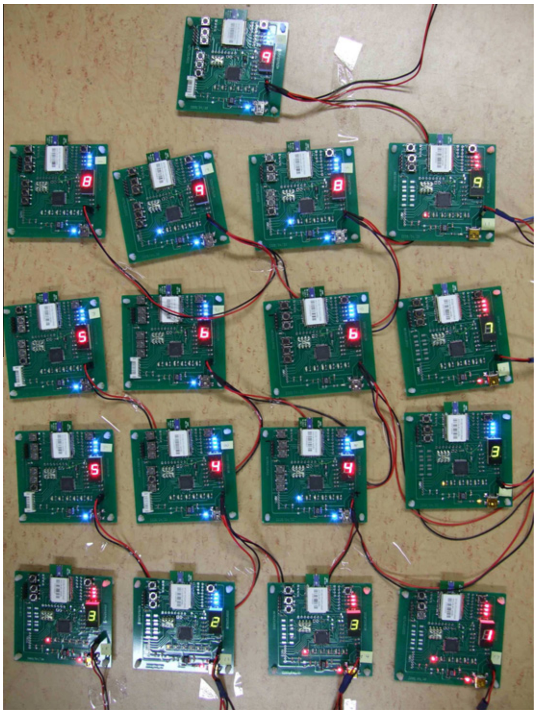


Fig. 15. A snapshot of DSM's behavior in an isolated 4 × 4 grid WSN.

Fig. 15 shows a snapshot after executing our rules, where the priorities of packets at nodes a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, and q are 9, 8, 9, 8, 9, 5, 6, 6, 7, 5, 4, 4, 3, 3, 2, 3, and 1, respectively. As it can be seen, packets are all in-order. Based on the implementation results, we conclude that our DSM is a lightweight protocol which can be easily implemented in a real sensor platform with a very small image in each microcontroller. It also demonstrates that DSM is quite suitable for a distributed WSN environment because DSM only needs local neighboring information.

7. Conclusions

We have addressed the distributed storage management and the opportunistic data collection problems in a field with multiple isolated and static WSNs, mules, and BSs by proposing two strategies called DSM and ODE.

Proof of correctness, simulation results, and prototyping experiences to demonstrate the feasibility of our results are presented. Extensive simulation results indicate that the data priorities collected by our DSM is very close to a global optimization and our ODE can collect fresher and fewer duplicated packets based on the designed utility model. It is shown that our DSM and ODE could collaborate well to collect important data among multiple isolated WSNs, mules, and BSs. Our prototyping results demonstrate that our proposed protocols are lightweight and could be deployed in a real distributed WSN easily.

We believe that our results have potential to be used in many WSN applications, such as those in outfields and back countries, as well as in many handset-based social gaming scenarios. While a small-scale prototype has been tested, we expect that our approaches can be tested in a larger environment and the results will be reported in our forthcoming work.

Acknowledgements

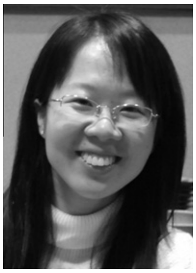
Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, by NSC grants 98-2219-E-009-019, 98-2219-E-009-005, and 99-2218-E-009-005, by AS-102-TP-A06 of Academia Sinica, by ITRI, Taiwan, and by D-Link. This work was also conducted under the “AMI Enhancement Project” of III, which is subsidized by MoEA, ROC.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks 52 (12) (2008) 2292–2330.
[2] I.F. Akyildiz, I.H. Kasimoglu, Wireless sensor and actor networks: research challenges, Ad Hoc Networks 2 (4) (2004) 351–367.
[3] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad Hoc Networks 3 (2005) 325–349.
[4] CITRIS, Design and Construction of a Wildfire Instrumentation System Using Networked Sensors. <http://firebug.sourceforge.net>.
[5] CENS, Terrestrial Ecology Observing Systems. <http://research.cens.ucla.edu>.
[6] C.-Y. Lin, W.-C. Peng, Y.-C. Tseng, Efficient in-network moving object tracking in wireless sensor networks, IEEE Transactions on Mobile Computing 5 (8) (2006) 1044–1056.
[7] A. Terzis, A. Anandarajah, K. Moore, I.-J. Wang, Slip surface localization in wireless sensor networks for landslide prediction, in: Proc. of ACM/IEEE Int'l Conference on Information Processing in Sensor Networks (IPSN), 2006, pp. 109–116.
[8] F.-J. Wu, C.-S. Huang, Y.-C. Tseng, My Tai-Chi book: a virtual-physical social network platform, in: Proc. of ACM/IEEE Int'l Conference on Information Processing in Sensor Networks (IPSN), 2010, pp. 428–429.
[9] L. Luo, C. Huang, T. Abdelzaher, J. Stankovic, Envirostore: A cooperative storage system for disconnected operation in sensor networks, in: Proc. of IEEE INFOCOM, 2007, pp. 1802–1810.
[10] R. Shah, S. Roy, S. Jain, W. Brunette, Data mules: Modeling a three-tier architecture for sparse sensor networks, in: Proc. of IEEE Workshop on Sensor Network Protocols and Applications (SNPA), 2003, pp. 30–41.
[11] H.-C. Lee, C.-J. Liu, J. Yang, J.-T. Huang, Y.-M. Fang, B.-J. Lee, C.-T. King, Using mobile wireless sensors for in-situ tracking of debris flows, in: Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys), 2008, pp. 407–408.
[12] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, P. Corke, Data collection, storage, and retrieval with an underwater sensor network, in: Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys), 2005, pp. 154–165.
[13] Y.-T. Huang, Y.-C. Chen, J.-H. Huang, L.-J. Chen, P. Huang, Yushannet: a delay-tolerant wireless sensor network for hiker tracking in Yushan national park, Mobile Data Management (2009) 379–380.
[14] Academia Sinica, YushanNet Project. <http://nrl.iis.sinica.edu.tw/YushanNet/index-2.php>.
[15] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 2001.
[16] G. Anastasi, M. Conti, E. Monaldi, A. Passarella, An adaptive data-transfer protocol for sensor networks with data mules, in: Proc. of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), 2007, pp. 1–8.
[17] A. Chakrabarti, A. Sabharwal, B. Aazhang, Using predictable observer mobility for power efficient design of sensor network, in: Proc. of ACM/IEEE Int'l Conference on Information Processing in Sensor Networks (IPSN), 2003, pp. 129–145.
[18] Y. Chen, W. Zhao, M. Ammar, E. Zegura, Hybrid routing in clustered dtns with message ferrying, in: Proc. of ACM/SIGMOBILE Workshop on Mobile Opportunistic Networking (MobiOpp), 2007, pp. 75–82.
[19] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proc. of ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2004, pp. 187–198.
[20] W. Zhao, M.H. Ammar, Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks, in: Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems (FTDCS), 2003, pp. 308–314.
[21] Y.-C. Wang, F.-J. Wu, Y.-C. Tseng, Mobility management algorithms and applications for mobile sensor networks, Wireless Communications and Mobile Computing 12 (1) (2012) 7–21.
[22] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: Proc. of IEEE INFOCOM, 2006, pp. 1–11.
[23] J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for DTNs, in: Proc. of IEEE INFOCOM, 2006, pp. 1–10.
[24] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, Lecture Notes in Computer Science 3126 (2004) 239–254.
[25] J.-H. Huang, S. Amjad, S. Mishra, Cenwits: a sensor-based loosely coupled search and rescue system using witnesses, in: Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys), 2005, pp. 180–191.
[26] Y. Wang, H. Wu, Delay/fault-tolerant mobile sensor network (DFT-MSN): a new paradigm for pervasive information gathering, IEEE Transactions on Mobile Computing 6 (9) (2007) 1021–1034.
[27] Visenti Pte Ltd., WaterWise Project in Singapore. <http://www.visenti.com/waterwise.html>.
[28] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi, in: Annual Conference of the IEEE Industrial Electronics Society (IECON), 2007, pp. 46–51.
[29] Efftronics Systems Pvt. Ltd., Automated Weather Station (Weather Monitoring unit). <http://www.made-from-india.com/showroom/efftronicssys/products/>.
[30] Jennic, Jennic – Wireless Microcontroller. <http://www.jennic.com>.

Yu-Chee Tseng got his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He is/was Professor (2000-present), Chairman (2005–2009), and Associate Dean (2007–present), Department of Computer Science, National Chiao-Tung University, Taiwan, and Chair Professor, Chung Yuan Christian University (2006–2010). He received Outstanding Research Award (National Science Council, 2001, 2003, and 2009), Best Paper Award (Int'l Conf. on Parallel Processing, 2003), Elite I. T. Award (2004), and Distinguished Alumnus Award (Ohio State University, 2005), and Y. Z. Hsu Scientific Paper Award (2009). His research interests include mobile computing, wireless communication, and parallel and distributed computing. He serves/served on the editorial boards for Telecommunication Systems (2005–present), IEEE Trans. on Vehicular Technology (2005–2009), IEEE Trans. on Mobile Computing (2006–present), and IEEE Trans. on Parallel and Distributed Systems (2008–present).

Fang-Jing Wu is a Research Fellow in Nanyang Technological University, Singapore. She received her B.S. degree in Mathematics from Fu Jen Catholic University Taiwan in 2001, M.S. degree in Computer Science and Information Engineering from National Chiao-Tung University Taiwan in 2004, and Ph.D. degree in Computer Science from National Chiao-Tung University, Taiwan in 2011. Her current research interests include cyber-physical systems, pervasive computing, and participatory sensing, sensor networks, and wireless and mobile networks.



Wan-Ting Lai, received the M.S. degree in Computer Science and Information Engineering from the National Chiao-Tung University, Taiwan. Her current research interests are primarily in wireless sensor networks.