

## Research Article

# Ambient Data Collection with Wireless Sensor Networks

Feng Wang,<sup>1</sup> Jiangchuan Liu,<sup>1</sup> and Limin Sun<sup>2</sup>

<sup>1</sup> School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6

<sup>2</sup> Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

Correspondence should be addressed to Feng Wang, fwa1@cs.sfu.ca

Received 9 February 2010; Accepted 7 June 2010

Academic Editor: Dan Wang

Copyright © 2010 Feng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the most important applications for wireless sensor networks (WSNs) is *Data Collection*, where sensing data are collected at sensor nodes and forwarded to a central base station for further processing. Since using battery powers and wireless communications, sensor nodes can be very small and easily attached at specified locations without disturbing surrounding environments. This makes WSN a competitive approach for data collection comparing with its wired counterpart. In this paper, we review recent advances in this research area. We first highlight the special features of data collection WSNs, by comparing with wired data collection network and other WSN applications. With these features in mind, we then discuss issues and prior solutions on the data gathering protocol design. Our discussion also covers different approaches for message dissemination, which is a critical component for network control and management and greatly affects the overall performance of a data collection WSN system.

## 1. Introduction

Wireless sensor networks have been applied to many applications since emergence [1]. Among them, one of the most important applications is data collection, where sensing data are continuously collected at each sensor node and forwarded through wireless communication to a central base station for further processing. In a WSN, each sensor node is powered by a battery and uses wireless communications. This results in the small size of a sensor node and makes it easy to be attached at any location with little disturbances to the surrounding environment. Such flexibility greatly eases the costs and efforts for deployment and maintenance and makes wireless sensor network a promising approach for data collection comparing with its wired counterpart. In fact, a wide range of real-world deployments have been witnessed in the past few years. Examples are across wildlife habitat monitoring [2], environmental research [3], volcano monitoring [4], civil engineering [5], and wildland fire forecast/detection [6], to name but a few.

The unique features of WSNs, however, also bring many new challenges. For instance, the lifetime of a sensor node is constrained by the battery attached on it, and the network lifetime in turn depends on the lifetime of sensor

nodes, thus, to further reduce the costs of maintenance and redeployment, the consideration of energy efficiency is often preferred in a WSN design. Moreover, these challenges are complicated by the wireless losses and collisions when sensor nodes communicate with each other.

In addition, the requirements specified by data collection applications also raise issues that need to be considered in the network design. First of all, to accurately acquire different types of data (such as temperature, light, and vibration), different sensors with different sampling rates may be deployed at different locations. Also as being relayed toward the base station, more and more sensing data will be accumulated along the delivery path. These issues may cause unbalanced energy consumptions over a WSN and significantly shorten the network lifetime if not handling carefully.

In this paper, we present a survey on recent advances of tackling these challenges. By comparing with both wired data collection networks and other applications of WSNs, we first highlight the special features of data collection in WSNs. With these features in mind, we then discuss issues and previous works on the data gathering protocol design. In addition, we discuss different approaches for message dissemination, which acts as an indispensable component

for network control and management and can greatly affect the overall performance of a WSN system for data collection.

The remainder of this paper is organized as follows. In Section 2, we compare WSNs for data collection with the wired data collection networks and WSNs for other applications, aiming to highlight the special features to be considered in the network design. Section 3 presents a detailed investigation on the data gathering protocol design and Section 4 discusses issues and prior solutions on message dissemination for network management and control. Finally, Section 5 concludes the paper and gives further discussions on the directions of future work.

## 2. Overview

**2.1. Wireless Sensor Networks.** As a type of newly emerged network, WSN has many special features comparing with traditional networks such as the Internet, wireless mesh network, and wireless mobile ad hoc network. First of all, a sensor node after deployed is expected to work for days, weeks, or even years without further interventions. Since it is powered by the attached battery, high efficient energy utilization is necessary, which is different from the Internet as well as wireless mesh and mobile ad hoc network, where either constant power sources are available or the expected lifetime is several order of magnitude lower than it is for WSNs.

Although a sensor node is expected to work through a long time, it is often not required to work all the time, that is, it senses ambient environment, processes and transmits the collected data; it then idles for a while until the next sensing-processing-transmitting cycle. To support fault tolerance, a location is often covered by several sensor nodes [7, 8]. To avoid duplicate sensing, while one node is performing the sensing-processing-transmitting cycle, other nodes are kept in the idle state. In these cases, energy consumption can be further reduced by letting the idle nodes turn to *dormant* state, where most of the components (e.g., the wireless radio, sensing component, and processing unit) in a sensor node are turned off. When the next cycle comes (indicated by some mechanism such as an internal timer), these components are then waken up back to the normal (*active*) state again. Define *duty cycle* as the ratio between active period and the full active/dormant period. A low duty-cycle WSN clearly enjoys a much longer lifetime for operation. This feature has been exploited in quite a few research works [9–12]. However, as will be shown later in this paper, the new working pattern also brings challenges to the network design.

Another special feature related to energy consumption is to control the transmission range of a sensor node. Previous researches have shown that one of the major energy costs in a sensor node comes from the wireless communication, where the main cost increases with the 2nd to 6th power of the transmission distance. As a result, the transmission range of a sensor node is often preferred to be adjustable and may be dynamically adjusted to achieve better performance and lower energy consumption.

**2.2. Data Collection.** In a data collection application, sensors are often deployed at locations specified by the application requirement to collect sensing data. The collected data are then forwarded back to a central base station for further processing. Traditionally, these sensors are connected by wires which are used for data transmission and power supply. However, the wired approach is found to need great efforts for deployment and maintenance. To avoid disturbing the ambient environment, the deployment of the wires has to be carefully designed. And a breakdown in any wire may make the whole network out of service and enormous time and efforts may be taken to find out and replace the broken line. In addition, the sensing environment itself may make the wired deployment and its maintenance very difficult, if not impossible. For example, the environment near a volcano [4] or a wildfire scene [6], where the hot gases and steams can damage a wire easily. Indeed, even in a less harsh environment like wild habitat [2, 3] or a building [13], the threats from rodents are still critical and make the protection of wires much more difficult than that of sensors. All these issues make wireless sensor network a pleasant choice as it emerges with technology advances.

On the other hand, although many research efforts have been done on WSNs, and quite a few prototype or preliminary systems have been deployed, data collection in WSNs is still in its early stage and its special features call for novel approaches and solutions different from other applications. For example, a common work pattern in most of other applications, such as target tracking [14], is that sensing data or information are locally processed and stored at some nodes and may be queried later by some other nodes. Data collection, nevertheless, requires all sensing data are correctly and accurately collected and forwarded to the base station, since the processing of these data needs global knowledge and is much more complex than that in other applications like target tracking. Thus, the major traffic in data collection is the reported data from each sensor to the base station. Such “many-to-one” traffic pattern, if not carefully handled, will cause high unbalanced and inefficient energy consumption in the whole network. As a concrete example, the energy hole problem, where sensor nodes close to the base station are depleted quickly due to traffic relay and create a hole shape area that leaves the remaining network disconnected from the base station, was reported and discussed in [15].

Unlike other WSNs, the sensors used in data collection are often in great amount and of different types, from traditional thermometer, hygrometer to very specialized accelerometer and strain sensor. These sensors work at their own sample rates specified by the applications, and the rates may be different from one to another, for example, a typical sampling rate of an accelerometer is 100 Hz, while the frequency to sample temperature is much lower. Such a difference in turn leads to different transmission rates to relay data from different type of sensors, which may further aggravate the unbalance of the traffic pattern and energy consumption and, thus, result in performance inefficiency.

In practice, after a data collection WSN is deployed, network setup/management and/or collection command

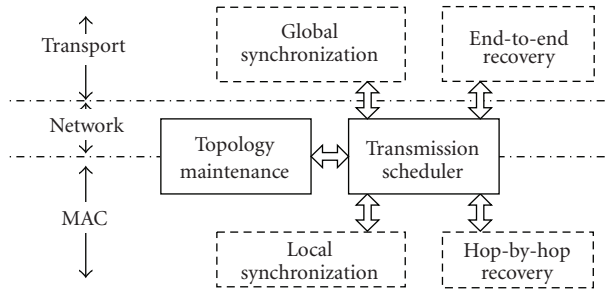


FIGURE 1: A generic architecture of a data gathering approach. Mandatory components are shown by solid squares and optional components are shown by dashed squares.

messages are disseminated from the base station to all sensor nodes by the message dissemination protocol. Then based on the information indicated by the disseminated messages, sensing data are gathered from different sensors and delivered to the base station through the data gathering protocol. It is worth noting that in a data collection system, the above process may work repeatedly, so that after one round of data collection, new setting/command messages are disseminated, thus, starting a new round of collection. In the following sections, we will investigate the designs of data gathering protocols as well as message dissemination approaches in detail on their recent progresses and discuss potential issues for future work.

### 3. Data Gathering Approaches and Issues

Data gathering approaches consider issues such as how to deliver sensing data from each sensor node to the base station. To achieve high efficiency, a cross-layer design is often involved, where the MAC, network, and transport layers are considered together to achieve multiple goals such as energy efficiency as well as reliability. Figure 1 illustrates a generic architecture for data gathering approaches. To collect data from sensor nodes, two mandatory components are topology maintenance and transmission scheduler. The topology maintenance component constructs a connected topology, often a tree rooted at the base station, and maintains the connectivity during network dynamics and link quality variations. The transmission scheduler then schedules packet transmissions based on information from other components so as to reduce collisions and energy costs. Given different QoS requirements such as throughput, latency, and reliability, different optional components may be added. Yet a more challenging issue is that sensor nodes are operating autonomously. Thus, the transmission scheduling algorithm needs to be designed to work in a distributed manner. In the following subsections, we will discuss recently proposed data gathering approaches categorized by the major QoS requirement being considered.

**3.1. Reliability.** One of the prior works [13] designed a data gathering approach with a stress on the reliability and proposed a hybrid scheme for reliable data delivery using

both hop-by-hop and end-to-end recoveries. Specifically, each node keeps tracking sequence numbers of packets it receives from a source node. A gap in the sequence numbers of received packets indicates packet loss. The sequence number of the missing packet and its source node ID are then stored in a missing list and piggybacked when a packet is forwarded. The node that previously relayed the missing packet will then schedule a retransmission when it overhears the piggybacked information. And to afford the retransmission in the hop-by-hop recovery, each newly received packet is cached for a short period. However, if heavy packet loss happens or the network topology changes due to dynamics such as link quality variations, the hop-by-hop recovery may fail due to the temporary overflow of missing lists or losing connections to previous forwarders. Thus, an end-to-end recovery scheme is necessary to such situations. In particular, if a node overhears a piggybacked missing list and finds some missing packets in the list sharing the same sources with those packets in its own packet cache, it then adds these packets into its own missing list and goes on to piggyback their information in its transmissions. By this means, missing packet information will trace back hop-by-hop until reaching the sources. The sources will then resend the packets and finish the circle of end-to-end recoveries.

**3.2. Latency.** Since wireless communications consume a significant portion of energy budget on sensor nodes, MAC protocols have been proposed to reduce idle listenings and turn the radio of the sensor node to sleep mode to save more energy. Such general designs, however, if being used for data collection without careful consideration, may introduce extra latencies and even more energy costs. For example, if the next-hop neighbor is still sleeping, a node has to wait some extra time (called *sleeping latency*) until the neighbor turns active. To reduce such sleeping latencies, one approach is to let a node overhear for possible transmissions so as to temporarily increase its active duration for potential incoming packets. However, this would make all nodes that overhear a transmission spend extra time being active and consume more energy while only several of them really participate in the traffic relaying.

To reduce sleeping latency as well as energy costs, the authors of [16] proposed DMAC to enhance data collection. The main idea is shown in Figure 2. Based on the network topology, sensor nodes along a delivery path from a source node to the base station will turn to receiving, sending, and sleep mode one after one in a sequential order. If there are more packets to send, a *More Data Flag* is piggybacked with each previous packet to indicate the next transmission. The receiver then turns back to receiving mode, instead of sleep mode, to listen to the following packet. For the case that a receiver has more than one sender, on receiving a packet from one sender, the receiver predicts that there are packets from other senders and turns to receiving mode. And if nothing is heard, it turns back to sleep mode. In addition, within a transmission time slot, CSMA is used for several senders to compete for one receiver, and another small time slot is reserved after each transmission slot for the failed sender to

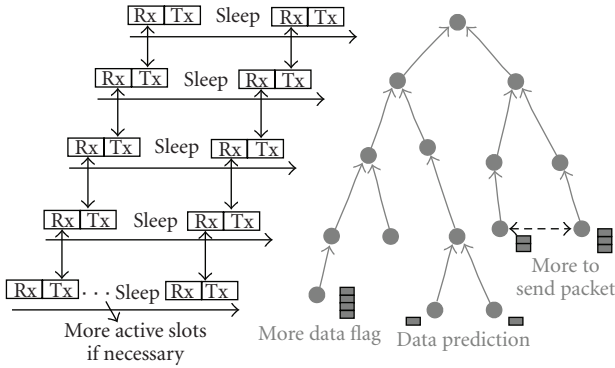


FIGURE 2: An illustration of DMAC [16].

send a small *More To Send* packet, so as to make the receiver listen to its retransmission instead of turning to sleep mode.

Another work proposed in [17] also targeted on minimizing latency and reducing energy costs. By assuming global synchronization, time slot is defined to be the duration for successfully transmitting a maximum transmission unit. Within one time unit, a sensor node can sleep to save energy or perform only one task of either sending or receiving. Given each sensor node has one packet to report to the base station during each round, for a linear topology as shown in Figure 3(a), one optimal schedule to minimize the time duration for one round of data collection is to let the even-level links and odd-level links be active alternatively, which is called *wavelike forwarding*. If there is any branch on the topology, as shown in Figure 3(b), the optimal schedule can be achieved by letting one path (e.g.,  $u_{t+k} \rightsquigarrow u_0$ ) conduct wavelike forwarding first, then after the branch ( $u_{t+k} \rightsquigarrow u_{t+1}$ ) of the path is finished, the remaining part together with the other branch ( $u_{t+k+r} \rightsquigarrow u_{t+k+1}$ ) will then form a new path and go on with wavelike forwarding. In general, for any tree topology, an optimal schedule can be achieved by recursively applying wavelike forwarding to each branch. Let  $N(u)$  denote the total number of nodes in the tree rooted at  $u$ . The authors showed that the time duration for all packets from the tree rooted at  $u$  to be forwarded up is  $2N(u) - 1$ . Furthermore, since the base station does not need to forward packets, it then can collect packets from two subtrees alternatively at the same time, for example, in Figure 3(c), if  $u_0$  is the base station, link  $u_1 u_0$  and  $u_{k+1} u_0$  can be active alternatively to send packets to  $u_0$ . Thus, the optimal schedule can be achieved by letting all the subtrees of the base station do wavelike forwarding simultaneously and the base station collects packets from its children alternatively in descending order of subtree size. The time duration for one round of data collection of the whole network is then derived as  $\max(2N(u_1) - 1, N(u_0) - 1)$ , where  $u_0$  is the base station and  $u_1$  is the child rooting the largest subtree.

**3.3. Throughput.** As the main traffic in a WSN for data collection is from all sensor nodes to the base station, the closer a sensor node is to the base station, the more packets it needs to relay. This will cause the funneling effect, as shown

in Figure 4, where the region close to the base station is heavily loaded and will experience significant collisions and packet losses if the MAC layer uses a CSMA-based protocol.

To solve this problem, Funneling-MAC [18] was proposed to improve the throughput of the network. The main idea is to adopt a TDMA protocol within the traffic intensity region (Figure 4), which is assumed within the coverage of the base station's transmission power. By monitoring the arriving traffic from each path within the region, the base station assigns time slots according to the traffic load. To keep synchronization, each time frame is started by a beacon from the base station, followed by the time slot assignment and then time slots for packet transmission. To facilitate emergency and control traffic, some time slots are reserved for transmissions by a CSMA protocol. In addition, the base station dynamically adjusts the size of the intensity region to exactly one hop smaller than the size that saturates all available time slots.

On the other hand, the authors of [19] focused on transport layer and proposed solutions to address congestion control and fairness issues. Different from wired networks and other wireless networks, the congestion control in that paper is done by a per-hop manner. Given a routing tree topology, each node measures its average rate  $r$  at which packets can be sent. Then this  $r$  is evenly divided by the number of sources in the descendants of the node (including itself). The result is then compared with the rate assigned by its parent and the smaller one is selected and broadcast to its children if no congestion happens. Otherwise, the selected rate is further decreased before being sent out. To achieve fairness, a node keeps the number of sources in the descendants of each child and uses these numbers as a weight to determine the packet from which child should be forwarded next. In addition, the paper proposed to use nonwork conservation for queues and showed that although at the cost of throughput, nonwork conservation helps to reduce the possibility of collisions and congestions.

**3.4. Energy Consumption.** Recently, the authors of [20] proposed an ultralow power data gathering scheme with a cross-design among MAC layer, topology control, routing, and scheduling. To achieve this, the scheme adopts a TDMA protocol, where a beacon is broadcast at the beginning of each round, allocating time slots to possible transmissions within this round. During tree topology construction and maintenance stage, nodes already integrated in the topology broadcast beacons and assign time slots for connection requests from remaining nodes. Nodes receive the beacons then send connection requests to one of the beacon senders and store the others locally for quick recovery when current connections fail. During data transmission stage, a parent node assigns each child a separate time slot for data reporting and local synchronization is achieved by letting all children listen to each beacon from the parent node. And by letting all nodes that are not listening or transmitting turn to sleep mode, a significant amount of energy can be saved. Besides, to resolve collisions, a pseudorandom delay jitter is introduced before a beacon is broadcast at each round.

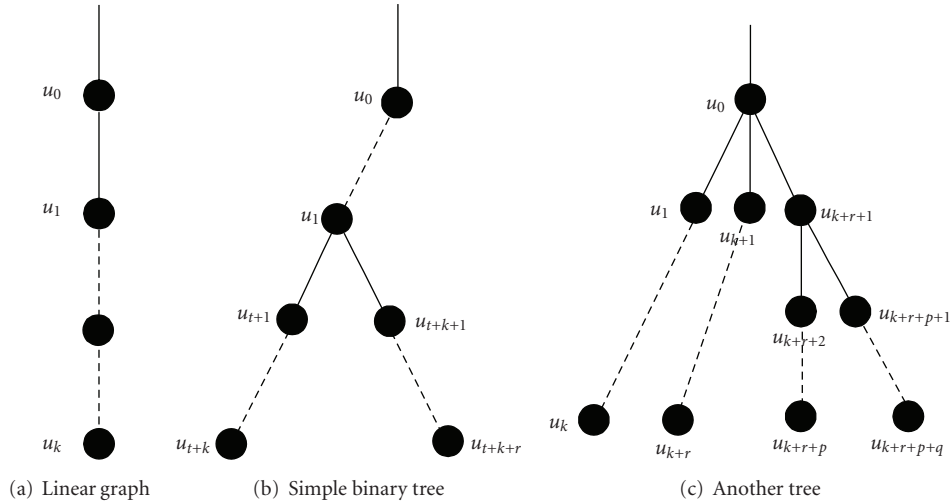


FIGURE 3: An illustration of time-optimal packet scheduling [17].

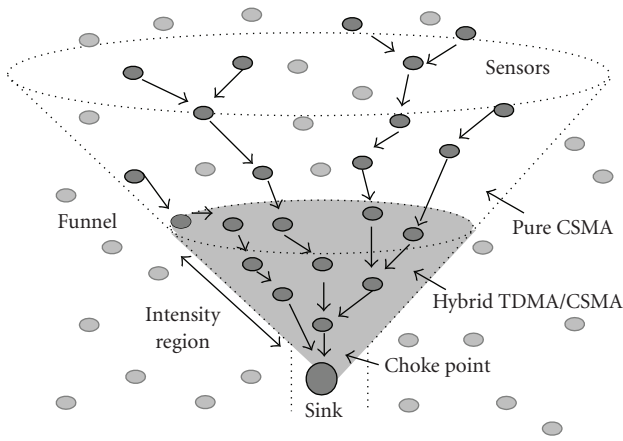


FIGURE 4: An illustration of funneling-MAC [18].

As mentioned earlier, when being relayed hop by hop towards the base station, sensing data will accumulate into a huge amount, which will quickly consume up the energy of those sensor nodes that relay these data traffics. Such situations may also happen in other WSN applications, where mobile base stations are proposed to proactively move within the sensing field to communicate with sensor nodes for traffic relaying [22]. However, due to the harshness of the sensing environment as well as to minimize the disturbances, such a solution is often unfeasible in the context of data collection.

To address this issue, the authors of [21] proposed a system in the scenario of data collection on high-rise structures such as skyscrapers and TV towers. In particular, instead of installing the base station on a fixed position, they proposed to put the base station on elevators used by high-rise structures. As the base station moves with the elevator, sensor nodes get opportunities to directly send data to the base station when it passes by, so that the traffic accumulation problem is alleviated. However, unlike mobile

base stations used in other applications, where the mobility can be well planned and controlled by the base station, the base station on an elevator can only passively move with the elevator. To fully exploit such passive mobilities, the authors proposed a solution with cross-layer design to jointly optimize link scheduling, packet routing, and end-to-end delivery, which would reduce the energy consumptions in two ways. One is during sensing data are routed to the base station, those sensor nodes holding the data will dynamically make a decision of whether to forward the data to the next hop closer to the base station so as to avoid excessive latency, or to wait for the base station moving close and transmit directly, which reduces the number of traffic relaying and, thus, the energy costs accordingly. The other way to save energy is by letting the base station periodically broadcast synchronization beacons while moving with the elevator, the whole network can be synchronized. The authors then design an algorithm to carefully schedule transmissions of the sensor nodes to eliminate possible wireless interferences caused by simultaneous transmissions, which greatly reduce the energy costs caused by wireless interferences and retransmissions. In addition, the proposed scheduling algorithm also considers the fairness and rate control among different source sensors.

**3.5. Summary.** We summarize the data gathering approaches discussed in this section in Table 1. Along these works, multiple QoS requirements can be considered jointly and the tradeoff among them can be further explored, which could be an interesting direction for future research. Also, although most of prior works assume a tree topology to be used for data collection, it has been noted that fault tolerance needs to be considered when deploying wireless sensor networks [23, 24], which often enrich the deployment topologies more than a tree structure with multiple paths provided. Thus, how to exploit such multipath features in the data gathering approach design to further enhance reliability is also available for exploring. Another issue is on energy saving

and extending the network lifetime. Most of the previous works depended on turning sensors into sleep mode to save energy and, thus, expect to extend the network lifetime. However, as mentioned previously, the “many-to-one” traffic pattern in data collection may cause high unbalance of energy consumption in the whole network and result in the premature termination of the network lifetime. To address this issue, the authors of [21] pose an interesting direction to exploit the special features of the sensing environment with little disturbances. Another interesting work is [3], where in the WSN system, the sunlight in the sensing environment is exploited to recharge the batteries of sensor nodes through the attached solar cells. Thus, how to exploit such special features and balance energy consumption to extend the network lifetime while still keeping good efficiencies is still an open question.

#### 4. Message Disseminations and Issues

Till now we have discussed how to collect data by WSNs, which follows the “many-to-one” traffic pattern. In such networks, however, there is another “one-to-many” traffic pattern where control messages are disseminated from the base station to all sensor nodes. Such traffic, although small in amount, is also critical to the system performance. Previous research works largely overlooked such traffic or assumed it can be easily solved by existing broadcast approaches from wired or other types of wireless networks. Nevertheless, the unique features of WSNs have shown necessity to call for novel solutions that can provide a network-wide broadcast service with both energy efficiency and reliability in this new context.

**4.1. Basic Flooding and Gossiping.** There have been numerous studies on broadcast in wired networks and in wireless ad hoc networks [25–27]. Among them, flooding and gossiping [1] are two commonly used broadcast approaches that can be easily adopted in WSNs. In flooding, each sensor node forwards the received message until the message reaches its maximum hop count. This approach provides high robustness against wireless communication losses and high reliability for message delivery. It, however, causes many duplicate messages being forwarded and, thus, leads to a significant amount of unnecessary energy consumptions. On the other hand, in gossiping, received messages are only forwarded with some predefined probability (In wired networks such as the Internet, gossiping was originally designed to let a received message be forwarded to a randomly selected neighboring node. Due to the broadcast nature of wireless communication, gossiping in WSNs is eventually evolved into the version mentioned above.). By theoretical analysis, a threshold probability exists to cover the whole network with high probability for a given topology and wireless communication loss rate. Thus, by setting the predefined probability just above the threshold, a great amount of duplicate messages can be avoided. Nevertheless, in practice, the predefined probability is very sensitive to the changes of the network topology and wireless communication loss,

which often leads to unsatisfactory reliability for message delivery.

Ideally, if without wireless communication loss, every sensor node needs to receive and forward the broadcast message at most once. Thus, though their basic forms are known inefficient, significant efforts have been made toward enhancing the efficiency of flooding or gossiping, while retaining their robustness in the presence of error-prone transmissions.

**4.2. Different Enhancements.** The author of [28] proposed a timing heuristic to reduce redundant message forwardings in the basic flooding as well as to extend the network lifetime. To suppress duplicate forwardings, a node only schedules a forwarding when it receives a broadcast message for the first time. Also a short latency named FDL (Forwarding-node Declaration Latency) is introduced before a node forwards a message, and if a forwarding for the same message is overheard, the node cancels its forwarding to further reduce duplicate forwardings. To extend the network lifetime, for a node  $u$ , its FDL is computed based on its residual energy  $E_t(u)$ , specifically, by the following equation:

$$\text{FDL}(u) = T \cdot \left(1 - \frac{E_t(u)}{E_{\text{ref}}(u)}\right) + t_D(u), \quad (1)$$

where  $T$  is a timing constant,  $t_D(u)$  is the maximum delay related to signal processing, transceiver switching and so forth at the potential forwarding nodes other than  $u$ , and  $E_{\text{ref}}$  is the maximum energy capacity of a battery. As a result, each time that several neighboring nodes receive a broadcast message, only the node with the highest residual energy and thus the shortest FDL will forward the message. Other nodes by overhearing will suppress their own forwardings to save the energy, so that the network lifetime is extended.

Smart Gossip [29], on the other hand, extended the basic gossip to minimize forwarding overhead while still keeping reasonable reliability. Different from the basic gossip that uses the same static forwarding probability for all sensor nodes, the authors proposed to dynamically adapt the forwarding probability on each node to its local topology and the originator of the broadcast message. Specifically, based on where the forwarded broadcast message comes from and who is its last forwarder, a node’s neighbors are divided into three sets, namely, *parent*, *child*, and *sibling*. The neighbors in the parent set are those that the node depends on to receive the first forwarded message; the neighbors in the child set are those that depend on the node to receive the first forwarded message; and the remaining neighbors are in the sibling set. Given an expected network delivery ratio  $\tau$ , the required per-hop delivery ratio  $\tau_{\text{hop}}$  can be estimated by the equation

$$\left(\tau_{\text{hop}}\right)^\delta = \tau, \quad (2)$$

where  $\delta$  is the estimated diameter of the network. Thus, for a node with  $K$  neighbors in its parent set, the required forwarding probability ( $p_{\text{required}}$ ) for each parent neighbor can be estimated using the equation

$$\left(1 - p_{\text{required}}\right)^K < \left(1 - \tau_{\text{hop}}\right). \quad (3)$$

TABLE 1: Different data gathering approaches.

Gathering approach	Network topology	Synchronization	Loss recovery	Congestion control	Sleep mode	Rate control	Source rate	Main QoS consideration
[13]	Tree	Not required	Hop-by-hop End-to-end	No	No	No	Any	Reliability
[16]	Tree	Local	Link layer	No	Yes	No	Any	Latency
[17]	Tree	Global	Link layer	No	Yes	No	Single	Latency
[18]	Any	Area covered by Base Station	Link layer	Partial	No	Partial	Any	Throughput
[19]	Tree	Not required	Link layer	Yes	No	Yes	Single	Throughput fairness
[20]	Tree	Local	Hop-by-hop	Yes	Yes	Yes	Single	Energy consumption
[21]	Any	Global	Hop-by-hop	Yes	No	Yes	Any	Energy consumption/ latency

Each node then collects  $p_{\text{required}}$  from all its child neighbors and uses the maximum as its own forwarding probability. Also, the three sets and  $p_{\text{required}}$  on each node are computed periodically based on recent message forwarding history, so as to make the forwarding probability adaptive to network dynamics (e.g., node failure).

A more recent work is RBP (Robust Broadcast Propagation) [30], which extended the flooding-based approach and targeted for high reliability broadcast. It lets each node immediately forward the broadcast message when the message is received for the first time. Then by overhearing, a node can quickly identify the percentage of its neighbors that have successfully received the message. Based on this percentage and the local density (the number of neighbors), a node determines whether to retransmit the message, where the principle is that for a low density, the message will be retransmitted until a high receiving percentage is achieved, while for a high density, a moderate percentage is enough. To counter wireless losses, explicit ACKs will be sent to nodes that are heard rebroadcasting a message several times. In addition, if a node finds itself highly dependent on another node to receive broadcast messages, the link between them is deemed as an important link. The downstream node will then notify its upstream node to increase the number of retransmissions to improve the probabilities of message deliveries.

To enhance reliability one step further, the authors of [31] proposed an approach with perfect broadcast reliability (i.e., all sensor nodes receive the broadcast message) for code redistribution and update propagation. To keep codes updated, each sensor node transmits a summary of its code if it has not heard a few other sensor nodes do so. When receiving a code summary from its neighbor, a node compares the received summary with its own. If the neighbor's summary is old, the node then sends its new code to the neighbor. And if the neighbor's summary is new, the node retransmits its own summary so as to trigger the neighbor to send the new code. Otherwise, a node counts the number of summaries received within one time interval,

if the number exceeds a threshold, the node suppresses its own transmission so as to save energy. And to balance energy costs, within each time interval, a node randomly picks its summary transmission time by following a uniform distribution. Moreover, the length of a time interval is set to a lower bound when a summary of new codes is received, so as to accelerate code updates. After that, the length of each next interval will be the double of the current one until it reaches to an upper bound, which further helps to reduce energy costs.

*4.3. Integrated with Duty Cycle.* The above approaches though are designed with different stress, such as reducing energy consumption or assuring high reliability, all make an implicit assumption that all network nodes are active during the broadcast process (referred to as *all-node-active assumption*). This assumption is valid for wired networks and for many conventional multihop wireless networks. It, however, may fail to capture the uniqueness of the energy-constrained applications in wireless sensor networks. In these applications, sensor nodes are often alternating between *dormant* and *active* states [9–12]; in the former, they go to sleep and thus consume little energy, while in the latter, they actively perform sensing tasks and communications, consuming significantly more energy (e.g., 56 mW for IEEE802.15.4 radio plus 6 to 15 mW for Atmel ATmega 128L microcontroller and possible sensing devices on a MicaZ mote). Define *duty cycle* as the ratio between active period and the full active/dormant period. A low duty-cycle WSN clearly has a much longer lifetime for operation, but breaks the all-node-active assumption. More importantly, the duty cycles are often optimized for a given application or deployment, and a broadcast service accommodating the schedules is thus expected for cross-layer optimization of the overall system.

To accommodate low duty cycles in WSNs, the authors of [32] proposed duty-cycle-aware broadcast, where a sensor node dynamically schedules message forwardings according to its neighbors' active/dormant patterns and whether they

TABLE 2: Different message dissemination mechanisms.

Dissemination mechanism	Basic approach	Balance energy consumption	Topology-aware	Support Duty-Cycle	Reliability	Delay cost	Message
[28]	Flooding	Yes	No	No	Moderate	Moderate	Very low
[29]	Gossiping	No	Yes	No	High	Low	Very low
[30]	Flooding	No	Yes	No	Very high	Low	Low
[31]	Flooding	Yes	No	No	Perfect	Moderate	Low
[32]	Flooding	No	No	Yes	Perfect	Moderate	Low
[33]	Flooding	No	Yes	Yes	Very high	Moderate	Low

have received the broadcast message. Specifically, each sensor node maintains the status of its neighbors within two hops (i.e., its neighbors and the neighbors of its neighbors). When in active state, a sensor node overhears message forwardings from other nodes and updates the status accordingly if some of its neighbors within two hops are within the range of an overheard message forwarding. The optimal message forwarding schedule within two hops is then computed by a dynamic programming algorithm based on the updated status. To handle wireless losses and guarantee the reliability, when receiving or overhearing a broadcast message, a sensor node adds the sender into a list and will only stop scheduling message forwardings after all the neighbors are added in the list. In addition, to accelerate the list updating and reduce the forwarding costs, a sensor node will piggyback its own list with its message forwardings and also update its own list according to the list piggybacked with a received or overheard broadcast message.

In some situations, the duty cycle of a WSN can become extremely low (e.g., %1 or less) such that a message forwarding from a sensor node is rarely received by multiple neighbors simultaneously [33]. Combined with the unreliable property of wireless links, this causes a sensor node has to forward the broadcast message many times before all its neighbors have received the message, and if not carefully handled, the total message costs can be extraordinarily large. To address this issue, the authors of [33] proposed to use energy-optimal trees for message disseminations, where an energy-optimal tree is constructed by starting from the source node and letting each sensor node selecting a neighbor as its parent if the neighbor has less hop distance to the source node and the best link quality to the sensor node. Since all sensor nodes work in extremely low duty cycle, simply depending on flooding along energy-optimal trees may introduce significant delays. The authors then proposed to use opportunistic forwardings through other unreliable links if a forwarding not scheduled on the energy-optimal tree can opportunistically cut off the delay of waiting for flooding through the energy-optimal tree. Also, to avoid the collisions and losses caused by multiple senders forwarding to one node simultaneously, a sensor node only selects the neighbors that can hear each other with high probabilities as the senders allowed to forward messages to it. And when forwarding a broadcast message, a small backoff delay based on the link quality is introduced so that the sender with the best link quality would transmit first and other nodes would

suppress their forwardings if they sense a forwarding has already been conducted.

*4.4. Summary.* The message dissemination mechanisms discussed in this section are summarized in Table 2. It is clear to see that although many mechanisms have been proposed, most of them did not consider the scenario of low duty-cycle WSNs except for the last two schemes. Along this new direction, many efforts are still required. First, theoretical models are expected to be introduced to more clearly understand how duty cycle and the active/dormant patterns would affect the message dissemination. Also, although the work proposed in [32] is designed to achieve perfect broadcast reliability, this is not mandatory in some scenarios, where it may be preferred to sacrifice a small portion of reliability so as to cut off more message costs. For such scenarios, a gossiping-based approach may be more favored for the system design. Moreover, in a low duty-cycle WSN, while the topology of active nodes changes frequently, the physical topology containing all nodes is relatively stable, which has been shown useful in [33] to build energy-optimal trees. Yet another interesting direction is to consider how to apply topology-aware techniques such as those used in [29, 30] to message disseminations and integrate the topology awareness seamlessly with the active/dormant patterns.

## 5. Conclusion

Wireless sensor networks have been applied to many applications since emerging. And data collection is one of the most important applications among them. In a data collection WSN, sensing data are continuously collected at each sensor node and forwarded through wireless communication to a central base station for further processing. This makes it different from other applications of WSNs as well as traditional data collection systems using wired networks. In this paper, we presented an in-depth survey on recent advances in the design issues and solutions for data collection systems using WSNs. Specifically, we first highlighted the special features of data collection in WSNs, by comparing it with both wired data collection network and other applications using WSNs. Bearing these features in mind, we discussed issues and solutions on the design of data gathering protocols. In addition, we discussed approaches for message disseminations, which are a critical component for network



control and management and thus also greatly affects the overall performance of data collection WSNs.

In the future, many issues still need to be further explored and possibly considered jointly so as to lead to a more efficient and long-lifetime data collection system. Some of the directions are to consider the special many-to-one traffic pattern for sensing data transmissions as well as the one-to-many traffic pattern for control/management message disseminations. Also, there may exist some special features (e.g., moving objects, sunlight, heat, and wind) in the sensing environment, which may provide new opportunities to enhance the system performance. For example, with the support of recharging devices that can collect energy from sunlight, heat, or wind, the residual energy distribution within the network system can be remarkably changed, which may lead to the designs of more efficient data gathering approaches and message dissemination mechanisms. Thus, how to dynamically exploit such features to effectively improve the system efficiency and lifetime while introducing little disturbances is an open question for further exploring. In addition, low duty cycle is considered as an effective way to extend the network lifetime of a WSN. Yet an interesting topic is to explore how its utilization in data collection WSNs interacts with other design issues.

## Acknowledgments

F. Wang and J. Liu's work is supported by a Canada NSERC Strategic Grant, an NSERC Discovery Grant, an NSERC DAS grant, and an MITACS Project Grant. L. Sun's work is supported by the State Key Program of the National Natural Science of China (Grant no. 60933011).

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 214–226, ACM, November 2004.
- [3] L. Selavo, A. Wood, Q. Cao, et al., "LUSTER: wireless sensor network for environmental research," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, ACM, 2007.
- [4] G. WernerAllen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proceedings of the 7th symposium on Operating Systems Design and Implementation (OSDI '06)*, USENIX, 2006.
- [5] S. Kim, S. Pakzad, D. Culler et al., "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 254–263, ACM/IEEE, April 2007.
- [6] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, vol. 2006, pp. 28–41, ACM, 2006.
- [7] H. Zhang and J. Hou, "On deriving the upper bound of  $\alpha$ -lifetime for large sensor networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MoBiHoc '04)*, pp. 121–132, ACM, May 2004.
- [8] H. Zhang and J. C. Hou, "Is deterministic deployment worse than random deployment for wireless sensor networks?" in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.
- [9] Y. Gu, J. Hwang, T. He, and D. H.-C. Du, " $\mu$ Sense: a unified asymmetric sensing coverage architecture for wireless sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS '07)*, IEEE, 2007.
- [10] J. Liu, F. Zhao, P. Cheung, and L. Guibas, "Apply geometric duality to energy-efficient non-local phenomenon awareness using sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 62–68, 2004.
- [11] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 28–39, ACM, November 2003.
- [12] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, ACM, November 2003.
- [13] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 13–24, ACM, November 2004.
- [14] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 129–143, ACM, October 2004.
- [15] S. Olariu and I. Stojmenović, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.
- [16] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 3091–3098, IEEE, April 2004.
- [17] W.-Z. Song, F. Yuan, and R. LaHusen, "Time-optimum packet scheduling for many-to-one routing in wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS '06)*, IEEE, 2006.
- [18] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 293–306, ACM, November 2006.
- [19] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 148–161, November 2004.
- [20] N. Burri, P. Von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 450–459, ACM/IEEE, April 2007.

- [21] F. Wang, D. Wang, and J. Liu, "EleSense: high-rise structure monitoring with wireless sensor networks," Tech. Rep., Simon Fraser University, 2010.
- [22] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.
- [23] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, "Complete optimal deployment patterns for full-coverage and  $k$ -connectivity ( $k \leq 6$ ) wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 401–410, ACM, May 2008.
- [24] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1649–1657, IEEE, May 2007.
- [25] S. Deering, *Scalable Multicast Routing Protocol*, Ph.D. dissertation, Stanford University, Stanford, Calif, USA, 1989.
- [26] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–803, 1997.
- [27] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '99)*, ACM, 1999.
- [28] X. Guo, "Broadcasting for network lifetime maximization in wireless sensor networks," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 352–358, October 2004.
- [29] P. Kyasanur, R. R. Choudhury, and I. Gupta, "Smart gossip: an adaptive gossip-based broadcasting service for sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS '06)*, pp. 91–100, IEEE, October 2006.
- [30] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza, "RBP: robust broadcast propagation in wireless networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 85–98, ACM, November 2006.
- [31] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04)*, 2004.
- [32] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 468–476, IEEE, April 2009.
- [33] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 133–144, ACM, September 2009.