Anycast Routing Protocol for Forest Monitoring in Rechargeable Wireless Sensor Networks

Demin Gao, Yunfei Liu, Fuquan Zhang, and Jun Song

College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China

Correspondence should be addressed to Demin Gao; gdmnj@163.com

Received 16 April 2013; Revised 7 November 2013; Accepted 14 November 2013

Academic Editor: Periklis Chatzimisios

Copyright © 2013 Demin Gao 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.

In rechargeable wireless sensor networks (R-WSNs), sensors have to adjust their duty cycles continuously owing to sporadic availability of energy and operate in a very low duty cycle because of current energy conversion technical limitations. Even though the network lifetime is not the main problem, packet delivery latency is critical because a sensor is in the dormant state most of the time in R-WSNs. These unique characteristics pose a new challenge for routing protocol design over traditional energy-static sensor networks. In this work, we introduce a routing protocol based on anycast communications technology, a novel design to minimize packet delivery latency for forest monitoring in R-WSNs. The key idea is to let a sender choose a closest neighbor to forward packets from forwarding candidates and an nearest sink node to accept the packet from all sinks. For multiple source nodes, we introduce an anycast technology based on Tabu search for each source establishing a minimal E2E delay routing path to reach anyone sinks. Through extensive simulation and experiments, we demonstrate that our anycast scheme based on Tabu search is efficient to provide smallest packets delivery latency in R-WSNs.

1. Introduction

Wireless sensor networks (WSNs), a promising technology, have been a topic of much interest to researchers due to their wide-ranging applications. For example, they have been used for military applications, environmental applications, health applications, and home applications [1, 2]. However, a fundamental problem in WSNs is the limited lifetime of sensor nodes. To this end, a significant amount of work has been carried out across the protocol stack to prolong the lifetime of WSNs. Examples of which include energy-efficient medium access control (MAC) protocols [3], duty-cycling strategies [4], energy-efficient routing [5], and topology control mechanisms [6]. Other examples can be found in [7, 8] and references therein.

An interesting approach to extend the lifetime of sensor nodes is the use of energy harvesting, which refers to harness energy from the environment or other source. Sensor nodes will equip them with rechargeable technologies that convert sources such as body heat [9], foot strike [10], and finger strokes [11] into electricity, such as a rechargeable battery or an ultracapacitor [12]. The energy from surrounding environment is free, unlimited, and environment friendly. Assuming energy neutral operation [13], a sensor node can operate perpetually if the harvested energy is used at an appropriate rate. Note that a harvesting node is said to achieve energy-neutral operation if the energy used is always less than the energy harvested, and the desired performance level can be supported in a given harvesting environment. The energy rechargeable device and energy storage device as the energy supplement components of R-WSNs replace the battery that is used in traditional WSNs (T-WSNs), which is the key difference between the new-fashioned WSNs and T-WSNs. The system architecture of an energy-powered sensor node is illustrated in Figure 1.

In this so-called energy harvesting or rechargeable WSNs (R-WSNs), although their lifetime is less of an issue, the available energy on nodes varies dramatically over time owing to the varying environment conditions. For example, a node that relies on the sun will extract more energy on sunny days as compared to when it is cloudy and extract no energy at night at all. To regulate energy consumption, nodes adapt their duty cycle according to available energy, which leads to high packet delivery latency or sleep delay [3]. Sleep latency is
the duration from the moment a packet is ready at the sender to the moment the destined receiver accepts the packet [14]. The key contributor to delay is the fact that if a node wants to communicate with a neighboring node, it has to wait for the corresponding neighbor to wake up. Sleep latency is usually in the order of seconds, which is much longer than other delivery latencies, such as processing delay, transmission delay, and propagation delay. The end-to-end (E2E) problem is more critical in R-WSNs than that in WSNs.

Anycast technology is utilized widely to solve the data transmission problems for multiple sinks deployed in WSNs. The paradigm of anycast communications [15, 16], also termed one-to-any communications, become very important to a network with multiple sinks: when a sensor node produces data, it has to send it to any neighbor node or sink available. A neighbor node or a sink selection strategy is to choose for each source a neighbor or a sink arbitrarily. An alternative strategy is to route data to the closest neighbor node or sink. Assuming that the sources and the sinks are uniformly distributed throughout the network, this simple strategy is assumed to improve data transmission efficiency and reduce latency.

In a nutshell, we propose a distributed anycast transmission scheme to reduce the E2E delay for sources-to-sink communications in R-WSNs. To the best of our knowledge, this is the first generic work that studies the use of anycast technology to reduce end-to-end delays in R-WSNs. The remainder of this paper is organized as follows. In Section 2, we present a number of existing anycast routing protocols and E2E latency guarantee solutions, while in Section 3 the motivation is provided. In Section 4, we present our method and design. Section 5 provides the analysis of our algorithm. Section 6 contains experimental results. Conclusions are presented in Section 7.

2. Related Work

Anycast is an important paradigm for a wireless sensor network in terms of resource, robustness, and efficiency for replicated service applications [17]. Routing protocols based upon anycast scheme are in accordance with the agreement [18], the extended model usually in the tree by hop count, physical interval, or time intervals for unit, to build an anycast tree and query is transported along the most fitting anycast tree [19]. The anycast scheme is utilized to cope with sink selection in routing protocol design for WSNs and improve the performance of routing protocol widely, whose ultimate goal is to prolong the network lifetime [20]. Other examples can be found in [21–23], and references therein. Even though multiple sinks are deployed in R-WSNs and sensor should also select one sink from the sink group to deliver packets, network lifetime is not a problem. It is impossible to rely on a fixed routing path for data transmission because of the dynamic topology attributing to dynamical energy supply in R-WSNs. So these routing protocols, which are designed for WSNs, do not adapt to R-WSNs. In particular, as one of the most important characteristics, low-duty cycle is not considered in routing design.

Various routing protocols have been proposed for low-duty-cycle sensor networks, for example, [24, 25]. Dynamic switch based forwarding (DSF) [26] is proposed for networks with possibly unreliable communication links and extremely low duty cycle, which utilizes forwarding sequence to reduce the time spent on transmitting a packet successfully at each hop in order to achieve expected communication delay and energy consumption. SCP [27] is designed for ultralow duty cycle (0.1%) sensor networks, which utilizes low power listening technique to poll a channel in order to save energy and reduce delay. Gu et al. [28] introduce two methods, including increasing duty cycle at individual node and place sinks to bound communication delay. These schemes encounter critical tradeoffs between network lifetime and data delivery latency. In R-WSNs, the lifetime is not the main problem which can be maximized by operating in an energy neutral mode for nodes.

By considering the dynamical energy supply for sensor nodes, Noh et al. introduce a duty-cycle-based low-latency geographic routing for asynchronous R-WSNs [29]. They propose D-APOLLO, an algorithm that periodically selects an appropriate duty cycle of each node to achieve minimum latency, which is based on the currently estimated energy by predicting energy consumption and energy expected from harvesting device. Sun and Xu [30] present an algorithm to reduce delivery delay that allows a sensor with packets to be sent dynamically select a forwarder from a forwarding sequence set, in which potential forwarders are sorted into the order of their wake-up time. Gu and He [31] introduce a method to increase duty cycle by strategically adding wake-up slots to nodes to reduce end-to-end delay to within a given bound. Gu et al. [32] further present energy synchronized communication (ESC) to synchronize the harvested energy at an individual node to minimize communication delays by shuffling and adjusting the working schedule of a node under different rates of energy harvest. However, more energy will be consumed, and collision will be serious in these protocols.

Different from earlier works, which either focus on static battery-powered network or minimize delay under energy constraints, in this work, we present a routing protocol based on anycast communications for forest monitoring in R-WSNs. Anycast scheme is used to resolve forwarding node selection for sensor node, including to choose an optimal neighbor node to forward packets from forwarding candidates and an optimal sink node to accept the packet from all sinks. In summary, on observing the lack of anycast technique’s consideration for reducing communication delay in existing
power management protocols. We introduce the first generic routing protocol algorithm with anycast scheme in R-WSNs. To the best of our knowledge, there was no prior work studying the problem of delay reduction using anycast.

3. Motivation

The motivation through the work is originated from our empirical experience of deploying rechargeable wireless sensor networks. In this work, we focus on using the harvested energy as the only energy source. The energy barely comes from the solar form. Therefore, the energy rate is generated only when there is sufficient sunlight or artificial light.

First, the dynamic duty cycle of a node is the most intuitive characteristics of R-WSNs. For studying the duty cycle of the node over time, we deploy a solar-powered sensor in an open field, as is shown in Figure 2. We assume the power consumption requirements of 83.1 μW in receive state and 76.2 μW in transmit state, which is presented in [33]. Assume the area of solar panel is 10 cm$^2$ to recharge energy for sensor nodes and the solar-energy conversion efficiency is about 10% [34, 35]. When a sensor node lies in direct sunlight at 12 am, the power density can reach 3700 μW/cm$^2$, energy harvesting rate is 320 mW, and the duty cycle achievable by the Crossbow MICAz can reach about 45%. However, when the sensor node lies indoor, the power density is about 320 μW/cm$^2$, the energy harvesting rate is 320 μW, and the duty cycle is estimated to be about 0.04% [33]. Therefore, the energy harvesting sensor networks are an environment-dependent network.

Second, the harvesting energy is still significantly lower than the power consumption for each wireless sensor node. Thus, node has to operate in a very low duty cycle which means that a node has to be activated shortly and stay at the sleep state most of the time in order to recharge. In our work, some experiments about the duty cycles of rechargeable sensor nodes were done. Figure 3 shows the affordable duty cycle of a node in three different weather conditions in spring. For example, a sensor CC2500 is used as the test device and lies outdoors. When the cloud area accounts for 1–10% of the sky, the duty cycles of a rechargeable node are ranged from 0.001% to 32% over 24 hours. From our empirical measurement results, a node can only operate about 3.5 hours if its duty cycle is over 20% using the total energy extracted from ambient environment over 24 hours. It is a long time for a packet reaching a sink. Therefore, quality routing protocol is critical for reducing transmission time in R-WSNs.

Third, in R-WSNs, it is unrealistic for a sensor node set fixed duty cycle to meet special requirements. Actually, the duty cycle of a node is adjusted passively according to the energy harvesting rate, which is significantly distinct from that in traditional WSNs that a sensor node can change its duty cycle freely only if it needs to do this. Even though various nodes perhaps lay in the same environment conditions, each node may harvest different amounts of energy due to some unpredictable factors, like a diverse solar panel angle. Therefore, the duty cycles of nodes may be different from each other. These characteristics have not been considered in T-WSNs, which will pose significantly challenges for routing protocol design in R-WSNs.

In R-WSNs, considering the dynamics energy supply, low duty cycle, and different harvested energy rate, the packet delivery latency is difficult to predicate and more critical than that of traditional WSNs, which affects the applications and popularization of R-WSNs significantly. In this work, we provide a generic and efficient anycast routing protocol for owning those characteristic’s sensor networks. To the best of our knowledge, few works have studied on the routing protocol for energy synchronization, low duty cycle, and duty cycle dynamics utilizing the anycast technologies.

4. Method and Design

4.1. Network Model. Consider a static rechargeable wireless sensor network modeled as an undirected graph $G = (V, A)$, where $V$ is the set of rechargeable sensor nodes and multiple sink nodes within the network. $A$ is the set of links, $A = \{A \mid (i, j) \in A, i, j \in V\}$. $G$ consists of a finite nonempty vertex set $V$ and edge set $A$ of ordered pairs of distinct vertices of $V$. At any point of time $t$, there are two states for
any sensor node with duty-cycle: active and dormant. In the active state, a sensor node can generate data after sensing its surrounding environment, transmit the data to its neighbors, or receive data from its neighbors. While a sensor node is in the dormant state, it turns off all its modules except for a timer to wake itself up.

The working time of all sensor nodes are divided into periodic working schedules, and every node shares its schedule with its neighbors. The working schedule consists of a set of active instances, during which a sensor node can transmit and receive packets. The transmitting and receiving activity happens in the active state. There are two situations for a dormant sensor node to wake up. The node has packets to transmit, or it is scheduled to the active state. In other words, a sensor node can wake up to transmit a packet at any time only if its receiver is in the active state but can receive a packet only when it is in the active state itself.

For all sensor nodes, since their neighboring nodes switch between active and dormant states regularly, the transmission between a pair of nodes becomes time-dependent strictly. In the paper, let the duration of periodic working schedules be \( t \) which can be divided into a sequence of time instances with length \( \tau \). The duration of \( \tau \) is the unit of working time for an activity. For a node, when it is in the active state within period time \( t \), its working schedules can be represented by a set of time instances, which contains one or multiple time instances with length \( \tau \).

For explaining the active and dormant activities utilizing the working schedule and time instance, let \( t_i \) denote the working schedule of the node \( i \) and \( t_i \) the time instance when node \( i \) is in the active state which can be called an active instance for node \( i \). Therefore, we can have \( t_i = \{t_{i1}, t_{i2}, \ldots , t_{in}\} \) for node \( i \) when it is in an active state for working schedule \( t_i \), for instance, a sensor node \( i \) with a period duration time \( 10\tau \) and working schedule \( t_i = \{2, 4, 7, 8\} \), as is shown in Figure 4.

In order to achieve minimal E2E delay from source node to a sink, minimal sleep latency in each one-hop transmission is expected to reach. Firstly, every node starts to broadcast its existence and working schedule to its two adjacent neighbors. Second, every node continues to broadcast its one neighbor’s working schedule to another neighbor node. Followed by neighbor discovery, every node knows the working schedules of adjacent neighbors and two-hop away node. Each node calculates the sleep latency between itself and its neighbor node and sends the result to its neighbor node. Finally, the sink node calculates the E2E delay from source node to the sink and nodes in the network start to execute delay maintenance process.

4.2. Description of E2E Delay. In the work, in order to calculate packet delivery latency or sleep delay, we assume that two sensor nodes are in a nodes group set \( \{i, i+1\} \), which means that the node with serial number of \( i + 1 \) is more closely than \( t \) to the sink. In the following parts, For the sake of clarity, node with the serial number of \( i \) is replaced by node \( i \). Assume that node \( i \) delivers packets at the time \( t_i \), which is one of active instances of working schedule of node \( i \). For adjacent node \( i + 1 \), the active instance \( t_{i+1} \) is one of the acts of working schedule. We assume \( t_{i+1} \) is the closest active instance to that of node \( i \), which means \( d_{ij+1}(t_{i+1}, t_i) \leq d_{ij+1}(t_{i+1} - t_i) \), \( i, n = 1, 2, \ldots , n, n \neq k \). The minimal delay from the node \( i \) to the node \( i + 1 \) is

\[
d_{ij+1}(t_{i+1}, t_i) = t_{i+1} - t_i. \tag{1}
\]

E2E Delay. For a packet ready at source node \( i \), the delay to reach the \( m \)-hop away sink node \( s \) is the sum of single-hop sleep latency. Consequently, it can be formulated as

\[
d_{is}(t_i, s) = \sum_{j=1}^{m} d_{ij+1}(t_i, t_{j+1}). \tag{2}
\]

From (2), we know that the E2E delay is affected by the working schedules of nodes and the number of hops from source node to a sink. Even though the working schedules of nodes are on the influence of energy supplement and hard to be revised, we can choose a short path from source to a sink. A short path is created based on nodes’ working schedules. In R-WSNs, a path with a minimal number of nodes is maybe not able to provide the smallest E2E delay. Therefore, when a path is established from source to the sink, minimal E2E delay (minimize \( d_{is}(t_i, s) \)) is prior to be considered. In our works, the main target for designing an opportunistic routing protocol is to minimize the E2E delay from source nodes to a sink.

4.3. Description of the Anycast Scheme. In the previous analysis, we know that the E2E delay is significantly affected by the working schedules of nodes and the number of hops while the number of hops to reach the sink is affected by the number of neighbor nodes of the sender, which is affected by the remaining energy of the sender. In the following section, we introduce the energy controlling scheme, neighbor node discovery, and an anycast scheme to achieve minimal E2E delay from all sensor nodes to a sink in R-WSNs. For the sake of clarity, we describe the scenario with a single sink node, which can be extended easily for scenarios with multiple sink nodes.

4.3.1. Energy Controlling Scheme and Neighbor Discovery. The assumptions, a node can wake up and transmit a packet at any time, which requires it has basic energy to support this suppose. In other words, a node should have enough energy to initiate a transmission with at least one neighbor at any-time. The energy needed to afford a transmission is different from each other, which depends on the distance between them. In this part, we introduce an energy controlling scheme and neighbor discovery based on energy controlling scheme.

In R-WSNs, we assume that \( E_i \) denotes the corresponding energy values for node \( i \) and \( E_{ij} \) denotes the energy values for node \( i \) and its neighbor node \( j \).
consumed when the node $i$ delivers packets to its nearest neighbor $i + 1$. For node $i$, whose energy $E_i \geq E_{i,j+1}$, should be kept at any time to ensure that it can be ready to wake up and send a packet to its neighbor node. $E_{i,j+1}$ denotes the energy consumed for node $i$ to wake up to receive a packet. Each node demands the same value of energy to wake up itself, which is not related to the distance.

If $\{t^*\}$ is one of the active instances of node $i$, its remaining energy $E_i$ should reach $E_{i,j+1} + E_{w}$ before the time $\{t^*\}$ in accordance with the previous assumption. In the work, we assume there are $k$ number nodes around a node $i$ and these nodes are in an ordered sequence $\{i+1, i+2, \ldots, i+k\}$, which means the node with serial number of $i+1$ is more closely than $i + 2$ to the node $i$ and the node with serial number of $i + k$ is the farthest away. If $E_i = E_{i,j+2}$, it means that the energy of node $i$ is enough to support it to communicate with node $i+2$ directly. In the process of operation, each node will monitor its remaining energy periodically.

To further demonstrate energy controlling scheme affecting the neighbor discovery, a simple example is provided, as shown in Figure 5. In the distribute network, the working schedule of node $i$ is set as $\{3, 8\}$ with the duration of 10 units of time, and three neighbors of node $i$ are in an ordered sequence $\{i+1, i+2, \ldots, i+k\}$. After joining the network, node $i$ initiates to recharge until its energy reaches $E_{i,j+1}$ in order to ensure that it can be ready to wake up at any time. Energy $E_{i,j+1}$ achieves at the time $t_1$, which does not belong to the working schedule of node $i$, and node $i$ continues to recharge, where $0 < t_1 < 3$, as it is shown in Figure 6.

We assume that node $i$ generates a packet at the time 2 and is expected to deliver the packet to its neighbor node; if $E_i < E_{i,j+1}$, node $i$ has to continue to recharge until it has enough energy to support the action; otherwise, the delivery cannot happen even if the node $i + 1$ is ready to receive the packet from node $i$. At the time 3, node $i$ has energy $E_{i,j+1} + E_{w}$ and is ready to send its packet to its neighbor node. However, node $i + 1$ is in the dormant state at the time 3 and impossible to receive any packets. At the time 4, the energy of node $i$ reaches $E_{i,j+1} + E_{w}$. Now, node $i$ has enough energy to deliver its packet to its two adjacent neighbor nodes $i + 1$ and $i + 3$, while only node $i + 3$ will be in active state and can receive packets from its neighbor node $i$ at the time 4.

Each node will establish a neighbor table based on energy controlling scheme, which contains potential receivers. The neighbor table is a stack, which holds adjacent neighbors of sender and the nearest node is put at the bottom of the stack. Figure 7 shows the neighbor table of node $i$ created over time. If the node $i$ has enough energy to communicate with its neighbor node $i+1$, node $i+1$ will be put in the neighbor table. When the energy of node $i$ reaches $E_{i,j+2} + E_{w}$, three nodes $i+1$, $i+3$, and $i+2$ are put into the table, which means the node $i$ has the ability to communicate with any node belonging to the neighbor table currently.

### 4.3.2. Neighbor Discovery Based on Anycast Scheme

In R-WSNs, the anycast technology is utilized to let one sensor choose another adjacent node from its neighbor table to accept packets. Therefore, the anycast technology can be used to resolve the one-to-any communication problems in R-WSNs. In this part, we will discuss how a sensor chooses one neighbor from its neighbor table and one sink from multiple destinations.

After a sensor node $i$ generates or accepts a packet at the time $t_1$, the node will check its remaining energy currently and establish neighbor table based energy controlling scheme. All its neighbors are in an ordered sequence $i + 1, i + 2, \ldots, i + k$. Now, the node $i$ starts to check which one node from its neighbor table will wake up to receive a packet at the time $t_1 + 1$. If there is no node being in an active state at the time $t_1 + 1$, node $i$ has to continue to wait for next time $t_1 + t, t \geq 2$ until there is at least one node being in the active state at the time $t_1 + t$. Assume that the time $t_1 + t$ belongs to the working schedules of $m, m \geq 1$ number of nodes, which means that all the $m$ number of nodes have the active instance $t_1 + t$ in their working schedule and will wake up at the time $t_1 + t; if m = 1$, the node $i$ will choose the neighbor node with active instance $t_1 + t$ as its next-hop node to receive the packet. If $m \geq 2$, there are more than two neighbors which can wake up to receive the packet, and the packet delay from
them to the node \( i \) is equal. The node \( i \) will choose a node with minimal sleep delay reaching a sink as the receiver.

To further demonstrate the selection method for getting a next-hop node from the neighbor table, we provide an example shown in Figure 8. We assume that a packet is generated at the time 5 from node \( i \), which has three neighbor nodes \( i+1, i+2, \) and \( i+3 \). Nodes \( i+2 \) and \( i+3 \) will be in the active state and have the ability to receive the packet at the time 5. However, the packet only needs to be forwarded to one node. The node with minimal sleep latency to a sink will be chosen as the next-hop node and will receive the packet from the sender. From Figure 8, node \( i+2 \) will accept the packet because it will reach sink \( s_1 \) after one point of time, while two points of time from node \( i+3 \) to the sink \( s_2 \).

In order to calculate the value of E2E sleep latency from a source node to a sink, we extend the neighbor-table form. The new neighbor table records not only the neighbors but also the E2E delay from the node to one sink. The recording delay in the new table is in the form of \((t^k_i, d_{ij}(t^k_i, s))\), where \( d_{ij}(t^k_i, s) \) denotes the minimal E2E sleep delay from the node \( i \) to the sink \( s \) when the node \( i \) receives a packet at the time \( t^k_i \), as is shown in Figure 9. If the node \( i+2 \) delivers its packet to the sink \( s_1 \), the sleep delay is 2 or 5. While the packet is delivered to the sink \( s_2 \), the new sleep delay is 3 or 5. So the minimal sleep delay is 2 from node \( i+2 \) to at least a sink. The recording delay \((5, 2)\) of the node \( i+2 \) represents that, if the node \( i+2 \) receives a packet at the time 5, the minimal sleep delay is 2 when the packet is transmitted to a sink.

After joining the networks, each sensor will calculate the E2E delay from itself to any one sink and store the value into the neighbor table. The delay calculation is initiated from the surrounding of a sink, firstly, and other far nodes will calculate the sleep delay to them according to their delay recording values. Consequently, each sensor node will establish a neighbor table containing all neighbor nodes and corresponding sleep delay. In the process of creating a neighbor table, a node does not master all sensors’ information except for adjacent neighbors.

4.4. The Anycast Transmission Scheme for Multiple Senders

4.4.1. The Anycast Technology for Multiple Senders. In R-WSNs, there are usually multiple senders deployed for collecting data, which poses a high challenge for routing protocol design. A receiver can only accept packet from one sender at a fixed point of time, which means if a sender accepts a packet from one sender at the time \( t \), it cannot accept any packet from other senders at the same time. Multiple senders have to contend with active instance of neighbor nodes. If a sensor has priority to use the active instance of one neighbor node, the active instance cannot be utilized by other sensor nodes owning the same neighbor.

An example is provided to demonstrate the anycast transmission scheme for multiple senders, as is shown in Figure 10. Node \( i \) and node \( i_1 \) have three neighbor nodes \( i+1, i+2, \) and \( i+3 \). Hypothesis, both of them have a packet to deliver to their neighbor node at the time 5. We assume that node \( i \) has the priority to choose next-hop node and selects the node \( i+2 \) to receive its packet at the time 5, which is occupied separately by the node \( i \). Now, only the node \( i+3 \) can be chosen by the node \( i_1 \) at the same time. The routing path for node \( i \) to a sink is \((i, i+1, i+4, s_1)\) and for node \( i_1 \) is \((i_1, i+3, i+5, s_2)\). Therefore, in the neighbor table of node \( i_1 \), the delay recording form \((5, 2)\) is removed, as is shown in Figure 11.

4.4.2. The Anycast Technology Based on Tabu Search for Multiple Senders. In this section, we propose a new anycast routing protocol scheme based on heuristic Tabu Search. The idea is to use Tabu Search method for all sources to achieve minimal E2E delay routing packets from the sources to any one sink in a distribute network. The Tabu Search is an advance local research method that uses rules and some mechanisms to lead toward the optimal solution. The Tabu research requires an initial solution and structure of neighbors. This method proceeds by moving from a solution...
to another. For each iteration, all the possible solutions are inspected, and the best one is selected.

After a sensor node generates or receives a packet, it will send the packet to any one of its neighbor nodes only if they are in active state, and the receiver continues to forward the packet to next node until the packet reaches a sink. In order to utilize the active instances of sensor nodes effectively and improve the transmission efficiency, the anycast technology based on Tabu search is utilized to achieve the best neighbor selection and sinks selection in R-WSNs.

The core of anycast routing protocol based on Tabu search for R-WSNs is to select a nearest sink as destination. The problem of optimal sink selection can be formulated as follows. Consider a case of $n$ sources and a group of $k$ sink nodes where $1 \leq k \leq n$. The problem is to assign the $n$ sources to $k$ sink nodes so that the total path sleep latency of the network is minimized. The problem can be formulated as a 0–1 integer programming problem as follows:

$$
\text{Minimum } f(x) \\
\text{Subject to } f(x) = \sum_{i=1}^{n} \sum_{j=1}^{k} \phi_{i,j} \lambda_{i,j},
$$

where $\phi_{i,j}$ is the path delay of the best route between node $i$ and node $j$, which is explained in (2), and $\lambda_{i,j}$ is a binary variable used for sink selection: if the best sink node chosen for node $i$ is the node $j$, then $\lambda_{i,j} = 1$; otherwise $\lambda_{i,j} = 0$.

The Anycast Technology Based on Tabu Search

(1) For a source node $i$, there are $m$ paths reaching at least a sink, $\text{PATHS} = \{x^0, x^1, \ldots, x^m\}$. $x^0$ is one of the routing paths from node $i$ to a sink. We assume that the path $x^0$ is the best routing $x^\text{best} = x^0$, which means that the E2E delay $f(x^0)$ is the minimal delay from node $i$ to a sink.

(2) When $\text{PATHS} \setminus \{x^\text{best}\} = \emptyset$, results are outputted, and calculation terminates. Otherwise, another path $x^k$, $k > 0$, is selected from $\text{PATHS}$, and the current path is $x^\text{now} = x^k$. If $f(x^\text{now}) < f(x^\text{now})$, $x^\text{best} = x^\text{now}$, $\text{PATHS} = \{x^\text{best}\}$, otherwise, $\text{PATHS} = \text{PATHS} \setminus x^k$, return 2.

Neighbor Node Selection. For a network with multiple source nodes, the source node with high priority has the priority to choose a neighbor to receive its packet, which is selected from its neighbor table using anycast technology. In the process of selection, only one node is chosen as long as it is in the active state when the sender delivers its packet. After the first source selection, the second node starts to choose its neighbor node and the process of node neighbor selection is repeated. The neighbor node selection has been discussed in the previous section.

Tabu List. Tabu list is an important part in Tabu search, and it prevents the search from short-term cycling. For a sensor node, its Tabu list contains the instances of its neighbors, which is empty initially and dynamical over time. Since a node only receives a packet from at most one sender at a point of time, it is impossible to accept any packet from other nodes simultaneously. Therefore, if an instance of a sensor node is utilized to receive a packet, the instance cannot be used another time at the same duration. The instances are removed from working schedule of sensor node and moved to Tabu list in order to avoid other nodes using the instance again. Other sensor nodes will not consider the active instances of neighbor nodes if they are in Tabu list when communicating with these nodes.

Tabu Length. Tabu length is the iteration number of a feasible solution that is forbidden. That is, for a certain number less than the Tabu length, the instance is not allowed for other applications if it is in the Tabu list. If the Tabu list is full, the instance that is the longest in the Tabu list is removed. In this paper, the Tabu length is set according to the requirements of a sensor node. If a sensor node decides to communicate with its neighbor node at the time $t$ for $m (m \geq 1)$ times, where $t$ is an active instance of the neighbor node, the active instance $t$ will be moved to Tabu list and Tabu length is set as $m$. After the sensor delivers a packet to the neighbor node one time, set $m = m - 1$. When $m = 0$, the active instance is removed from Tabu list, and other nodes can communicate with the neighbor node at the time $t$ again.

The program is stopped when its iteration number exceeds the limit. Similarly, another purpose is to limit the occurrence number of a solution or an objective value. The program will also terminate when the related numbers exceed the corresponding limit. When a sensor node wants to stop transmission even if $m \geq 1$, it can send controlling packet to inform its neighbor node that the transmission finished and let its neighbor node remove the active instance from Tabu list. The Tabu list is updated timely, and new active instances are moved to Tabu list or removed from Tabu list over time constantly.

In order to further demonstrate the anycast transmission scheme based on Tabu search for multiple senders, we analyze transmission from node $i$ as shown in Figure 10. In Figure 12, three neighbor tables of node $i + 1$, $i + 2$, and $i + 3$ are established; other neighbor tables of node $i$, $i_1$ have been
shown in Figures 9 and 11. We assume the node $i$ generates or receives a packet at the time 4. If the node $i$ has the priority to establish a minimal delay routing path to reach a sink node, the anycast transmission scheme based on Tabu search is executed as follows. 

(1) There are at least four paths reaching at least a sink node. Consider $\text{PATHS} = \{x^0 = (i, i+1, i+2, i+4, s_j), x^1 = (i, i+1, i+2, i+5, s_j), x^2 = (i, i+2, i+4, s_j), \text{and } x^3 = (i, i+2, i+5, s_j)\}$. The node $i$ selects a path from $\text{PATHS}$ randomly, which is $x^0 = (i, i+1, i+2, i+4, s_j)$ and the E2E delay $f(x^0) = 16$, where $x^\text{best} = x^0$, $\text{PATHS} = \{x^1, x^2, x^3\}$.

(2) Since $\text{PATHS} \setminus \{x^\text{best}\} \neq \emptyset$, $x^\text{now} = x^1$, the E2E delay $f(x^1) = 16$, and $f(x^\text{now}) = f(x^\text{best})$, $x^\text{best} = x^0$, $\text{PATHS} = \{x^2, x^3\}$. The iteration continue and $x^\text{now} = x^2$, the E2E delay $f(x^2) = 3$, and $f(x^\text{now}) < f(x^\text{best})$; therefore, $x^\text{best} = x^2$, $\text{PATHS} = \{x^3\}$. When $x^\text{now} = x^3$, the E2E delay is $f(x^3) = 4$ and $f(x^\text{now}) > f(x^\text{best})$; therefore, $x^\text{best} = x^3$, $\text{PATHS} = \emptyset$. The program termination and the best routing path are $x^\text{best} = (i, i+2, i+4, s_j)$ and the minimal E2E delay $f(x^\text{best}) = 3$ from node $i$ to the sink $s_1$.

The best routing path is outputted after the program termination of node $i$. While the active instances used in the path are moved to Tabu list, now, we assume node $i_1$ has second priority to create its minimal delay routing path and ready to deliver a packet at the time 3. The anycast technology based on Tabu search is executed and the process of routing path discovery is executed again. Since the active instances 5 of node $i+2$ and 6 of node $i+4$ are in Tabu list, a best path $x^\text{best} = (i_1, i+3, i+5, s_j)$ is selected and the minimal E2E delay is $f(x^\text{best}) = 5$ from node $i_1$ to the sink $s_2$. Assume that node $i_1$ finishes packet delivery and releases the path $x^\text{best}$. So the active instances in the path $x^\text{best}$ are removed from Tabu list and can be used to execute other tasks at the point of active instances. Now, node $i_1$ will establish a new routing path to reach a sink node depending on new active instances appearing in neighbor nodes. The optimal path $x^\text{best} = (i_1, i+2, i+4, s_j)$ is selected and the new E2E delay is $f(x^\text{best}) = 4$ from node $i_1$ to another sink $s_2$. Since new E2E delay declines, packets will be delivered along the new route, which will replace the original path.

The anycast technology based on Tabu search can achieve three purposes for multiple source nodes in R-WSNs. Firstly, the anycast scheme can be utilized to achieve optimal neighbor’s and sink’s selection. Secondly, a sensor node can establish a routing path with minimal E2E delay to reach at least a sink node. Third, in the process of creating a path, a sensor does not need to know which sink will receive the packet before the transmission finished, which means that the sink selection is transparent for sensor nodes. At the same time, anycast technology based on Tabu search can release the pressure of transmission for sinks effectively, which benefits from multiple sinks sharing the packets’ transmission.

5. Analysis

5.1. Experiment Setup. In the simulation, up to 300 nodes are randomly deployed in a 500 m * 500 m square field. The maximum communication range of each node is set to be 50 m. All nodes’ energy devices are rechargeable and transmission powers are adjustable. Every two nodes can communicate with each other directly within their transmission range. Except some cases, the default number of nodes in the network, node duty cycle, and delay bound is 300, 2%, and 250 units of time, respectively. Every data point in simulation figures is obtained by averaging 50 runs with different random seeds, node deployment, and node working schedules.

What we use consists of a solar panel optimized for outdoor use, two eZ430-RF2500T target boards and one AAA battery pack, which is rechargeable and can be recharged repeatedly. The target board comprises the TIMSP430 microcontroller, CC2500 radio transceiver, and an on-board antenna. The CC2500 radio transceiver operates in the 2.4 GHz band with data rate of 250 kbps and is designed for low power wireless applications. The harvested energy is stored in EnerChip, a thin-film rechargeable energy storage device with low self-discharge manufactured by Cymbet, as is shown in Figure 13.

Our projects get the support of government funds and Provincial Science and Technology Center (PSTC). We have deployed a large number of sensor nodes and monitoring system in the Zilin Mountain and Nanjing Forestry University (Nanjing City, China) (http://njfu00.h006.996h.cn/).
the Internet site is valid from 2013-10 to 2014-10), which is used to collect various sensory data, which contains temperature, humidity, and illumination from a forest, as shown in Figure 14. One of our objectives is to maintain the attainability of the network for as long as possible and collect accurate and as much as possible data to support the research about forest, which is also demonstrated to be necessary for other applications of sensor networks. Hundreds of sensor nodes and multiple sinks are deployed in the two places to monitor the coverage area. Most of them are powered by two energy systems, including a rechargeable battery and a common battery together.

5.2. System Performance Comparison. In order to understand the performance of our Routing Protocol for Anycast Communications (RPAC) under network settings; in this section, we provide an algorithm for performance comparison, which is proposed by Eu et al. [36]. In [36], an opportunistic routing protocol (EHOR) for multihop WSN-HEAP is introduced to resolve the optimal forwarder problem, which uses a region-based approach by grouping nodes together to reduce delay. We first compare the per-hop delay between our algorithm and EHOR under distinct duty cycles and number of sensors, as is shown in Figure 15. From Figure 15, we can know that the average delay per-hop in both algorithms decreases with the duty cycle increasing in three different scenarios, and the average delay of PRAC is obviously lower than the EHOR.

In scenario 1, up to 150 sensor nodes and 2 sinks are deployed. From Figure 15(a), we know that the delay per-hop of EHOR is always greater than that of our algorithm. In particular, when the duty cycle of a node decreases greatly, the delay of EHOR increases more quickly than that of the PRAC. From Figure 15(a), we can see if the duty cycle of a node is lower than 1%, the delay per-hop of sensor is lower about 5% than that of EHOR. In other words, when sensor nodes keep in low duty cycles, our algorithm will operate more efficiently than EHOR. Therefore, our algorithm is more stable than EHOR in delay performance. In the EHOR, all sensor nodes are divided into multiple regains based upon the distance to the sender within sensor transmission range. Only one node will be chosen as the forwarding node in each region, which depends on the remaining energy and the distance to the sink. Even though the EHOR can achieve to maximize goodput, E2E delay is as secondary performance metrics. In the process of transmission, the transmission power controlling technology has not been considered for each sensor node. Therefore, a sensor node cannot adjust its neighbor table timely, which cause the E2E delay to be higher than RPAC.

In scenario 2, 300 sensor nodes and 2 sinks are deployed. The nodes density increases significantly. From Figure 15(b), we know that the delay per-hop decreases greatly for both algorithms because a sensor node will have more neighbor nodes to receive its packets. When density of sensor nodes increases, since more sensor nodes will be contained within the neighbor table for each sensor node, it is easier for a sensor node to find a receiver when it wants to deliver a packet. For different duty cycles, the delay per-hop in our algorithm is always lower than that of EHOR. In the third scenario, we deploy more sinks in the monitoring field. Even though sensor node’s density does not increases, the number of nodes lying in the path from source node to sink decreases. Therefore, the E2E delay that a packet is delivered from a source node to sink decreases. 300 sensor nodes and 4 sinks are deployed in scenario 3; the result is shown in Figure 15(c), the data tests and verifies our analysis.

Figures 15(d), 15(e), and 15(f) show the delay per-hop of the network under a different number of nodes when the duty cycles of sensor nodes are 1%, 3%, and 5%. From the three diagrams, we see that the delay per-hop decreases when the number of node’s increases for both algorithms. If we improve the duty cycles of sensor nodes, the average delay per-hop of sensor’s further declines. High duty cycles will cause the packet’s delivery faster than before because sensor nodes will be in the active state more time to listen to the channel and ready to receive a packet. The figures illustrate that the delay per-hop of our algorithm is a little lower than that of EHOR for every different number of nodes and duty cycles.

The data generation rate of each sensor node is easy to be controlled, which depends upon the requirement of application. In our monitoring system, for an event-monitoring system, sensor will generate a data when it detects an occurrence of event or the timer reaches point in time. In fact, sensor node generates packets periodically or intermittently. Furthermore, the frequency of the data generation differs from one application to another. In our evaluation, we
increase the data generation rate from 5% to 35% greatly in order to further understand the performance of our routing protocol and EHOR.

In the applications for real-time monitoring requirement, minimum transmission latency is demanded to ensure that the data are forwarded to destination as soon as possible. Therefore, we evaluate the average delay per-hop with the distinct number of nodes and data generation rate in the network. The result in Figure 17 shows the average delay per-hop as the data generation rate rises under a different number of sensor nodes and duty cycles of sensors in each protocol. Since the transmitting ratio rises substantially when a large number of packets are generated accumulated in sensor nodes; hence, the packet delivery delay rises significantly when we improve the data generate rate over time.

Figures 16(a), 16(b), and 16(c) show the average delay varying data generate rate under a different number of sensor nodes and sinks, respectively. From the three diagrams, we can know that the delay in EHOR is larger than that in our protocol in every data generate rate. In particular, the advantage of our algorithm is more notable when a large volume of data is generated. This is because packets reach the sink node after multiple retransmission by both protocol. The delay of our protocol is a slight lower than that of EHOR. The reason can be attributed to the more adaptable transmission scheme, since a sensor node can adjust its transmission power to control transmission range freely according to its remaining energy. In our protocol, more packets from nodes far away from the sink can reach the sink node because the source node can choose its neighbor node more flexible than that in EHOR.

Figures 16(d), 16(e), and 16(f) show the average delay under data generate rate increasing when the duty cycle of nodes is 1%, 3%, and 5%. From the diagrams, we see the delay per-hop increasing when we improve data generate rate from all three different duty cycles of sensors. When the duty cycles of sensor nodes increased, the average delay decreases significantly. High duty cycle means that a sensor node is in the active state more times and has more opportunities to receive or transmit a packet. Therefore, a packet is transmitted to the sink more quickly than before due to more neighbor nodes being ready to forward the packet. The delay of EHOR is a little higher than that of our protocol. For example, when the average duty cycle of a sensor node is 5% and the data generate rate is 30%, the delay of EHOR is about 2% taller than that of our algorithm.

6. System Implementation

6.1. Experiment Setup. In addition to large-scale network simulations, we deploy 16 nodes and a sink in a distribute network to further explain the energy of sensor node changing over time. Packet is generated from a source node and will be transmitted to the sink node. Other nodes can forward the packets from source node to the sink. After deployment, every node initiates to generate working schedule randomly.
with 20% duty cycle, which means every node can generate only average two active instances within each period time. Each sensor is equipped with a rechargeable battery, which owns the same storage capacity. When a sensor node joins the network, it has to recharge itself and let its remaining energy beyond a certain value to ensure that it can transmit a packet to its neighbor at any time.

6.2. System Insights. After being deployed in the monitoring area, every node initiates to broadcast its existence and working schedule to its adjacent neighbors. Followed by neighbor discovery, every node knows the working schedules of neighbors. When a packet is generated from the source node, the packet is forwarded from the source node to the sink node utilizing anycast transmission scheme based on Tabu search. The neighbor node selection is on the influence of remaining energy of the sensor node. To further illustrate the remaining energy changing over time, we monitor the battery of sensor in the process of packet delivery. Remaining energy of all sensor nodes, including 16 numbers of sensors in period times, are shown in Figure 17.

In our experiments, one unit of time equals about 4 ms, maximum storage capacity of a temporary battery is about 300 \( \mu \)W, and surplus energy will be stored in another battery. Figure 17(a) shows the remaining energy of a sensor whose ID is 1 in a cycle period of time. From the time 0 to 400, node 1 recharges itself and the energy increases substantially reaching about 200 \( \mu \)W. At the time of 400, an event occurs within the monitoring area. The sensor node generates a packet to record the event and is ready to transmit to neighbor nodes. Hence, the energy decreases from the time 400 to 500. The sensor node continues to recharge itself from time 500 to 600. At the time 600, the sensor node sends the packet to one of its neighbor when the neighbor node wakes up by checking its neighbor table. The energy of sensor node 1 decreases significantly at the time 600 for packet transmission. After the delivery, the sensor node has not received any feedback from its neighbor node. Therefore, it retransmits the packet at the time 750 and the energy reduces to 75 \( \mu \)W. Now, the packet is transmitted from the sensor node to its neighbor node and is stored into the data buffer.

The energy of the sensor whose ID is 3 over a period unit time is shown in Figure 17(c). The energy of node 3 reaches about 180 when the time is 200. Then, a packet coming from its neighbor and node 3 wakes up itself to receive the packet and stores the packet in its data buffer. The energy of node 3 decreases to 75 after receiving the packet. From the time 250 to 550, node 3 continues to recharge itself and has the energy about 150 at the time 550. At the same time, the sensor listens to the channel to check whether there is a packet coming from its neighbors utilizing weak sensing device in order to save energy. Even though there are no packets being in the channel, the sensor has wasted a little energy for channel listening. The energy of sensor node 3 reaches about 270 at the time 700. When node 3 finds that one neighbor will wake up at the time 700 by checking its neighbor table, node 3

![Figure 16: Delay per-hop of sensor node under packet generation rate and node density.](image-url)
listens to the channel and is ready to send the packet to its neighbor. If the channel is idle, the packet will be forwarded to its neighbor from node 3.

7. Conclusion

In this work, we first define the network model and elaborate on the packet delivery process. The unique characteristics of R-WSNs cause that packet delivery latency is critical. To improve data transmission efficiency, we introduce a routing protocol based on anycast communications technology, a novel design to minimize packet delivery latency from source node to the sink for forest monitoring in R-WSNs, which is particularly designed for low-duty-cycle and dynamical duty-cycle wireless sensor networks. To our knowledge, this is the first work on routing protocol that utilizes anycast scheme and Tabu search to establish minimal E2E delay paths. In our design, each sender chooses a nearest neighbor to forward packets from forwarding candidates and a closest sink node to accept the packet utilizing anycast scheme.

For multiple source nodes, we provide an anycast technology based on Tabu search for each source node establishing a routing path with minimal E2E delay to reach at least a sink. For establishing the smallest delay path, we propose a mechanism of the heuristic Tabu search solution. Through extensive simulation and experiments, we demonstrate our anycast scheme based on Tabu search approaches the optimal performance and achieves a shorter average with less than about 8% latency over traditional method in various network settings. The anycast routing protocol based on Tabu
search proposed in the paper is designed for the scenario where the working schedules of sensor nodes are fixed. In the future, we shall extend our work into the scenario where working schedules of sensor can be adjusted flexibly to provide better performance.

Conflict of Interests
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments
This work was supported in part by The Six Talent of Jiangsu Province under Grant DZXZ-149-148. The authors also received two projects funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions and the Research of Modern Educational Technology of Jiangsu Province under Grant 2013-R-26936.

References
[1] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, “Sensor information networking architecture and applications,” IEEE Personal Communications, vol. 8, no. 4, pp. 52–59, 2001.
[2] E. Shi and A. Perrig, “Designing secure sensor networks,” IEEE Wireless Communications, vol. II, no. 6, pp. 38–43, 2004.
[3] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys ’03), vol. 11, pp. 171–180, November 2003.
[4] Y. Gu and T. He, “Data forwarding in extremely low duty-cycles sensor networks with unreliable communication links,” in Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’07), pp. 321–334, November 2007.
[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS ’00), p. 223, January 2000.
[6] S. Kang, S. Lee, S. Ahn, and S. An, “Energy efficient topology control based on sociological cluster in wireless sensor networks,” KSII Transactions on Internet and Information Systems, vol. 6, no. 1, pp. 361–380, 2012.
[7] D. Koutsonikolas, S. M. Das, Y. C. Hu, and I. Stoimenovic, “Hierarchical geographic multicast routing for wireless sensor networks,” Wireless Networks, vol. 16, no. 2, pp. 449–466, 2010.
[8] N. Dimokas, D. Katsaros, and Y. Manolopoulos, “Energy-efficient distributed clustering in wireless sensor networks,” Journal of Parallel and Distributed Computing, vol. 70, no. 4, pp. 371–383, 2010.
[9] T. Starner, “Human-powered wearable computing,” IBM Systems Journal, vol. 35, no. 3–4, pp. 618–629, 1996.
[10] N. S. Shenck and J. A. Paradiso, “Energy scavenging with shoe-mounted piezoelectrics,” IEEE Micro, vol. 21, no. 3, pp. 30–42, 2001.
[11] J. Kymissis, C. Kendall, J. Paradiso, and N. Gershenfeld, “Parasitic power harvesting in shoes,” in Proceedings of the 2nd International Symposium on Wearable Computers, pp. 132–139, October 1998.
[12] H. El Ghor, M. Chetto, and R. Hage Chehade, “A nonclairvoyant real-time scheduler for ambient energy harvesting sensors,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 732652, 11 pages, 2013.
[13] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” ACM Transactions on Embedded Computing Systems, vol. 6, no. 4, article 32, 2007.
[14] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’02), pp. 1567–1576, June 2002.
[15] W. Hu, N. Bulusu, and S. Jha, “A communication paradigm for hybrid sensor/actuator networks,” International Journal of Wireless Information Networks, vol. 12, no. 1, pp. 47–59, 2005.
[16] N. Thepvilojanapong, Y. Tobe, and K. Sezaki, “HAR: hierarchy-based anycast routing protocol for wireless sensor networks,” in Proceedings of the 5th Symposium on Applications and the Internet (SAINT ’05), pp. 204–212, February 2005.
[17] N. Y. Ithaca, “Towards a global IP anycast service,” in Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’05), pp. 301–312, 2005.
[18] M. Koziuik and J. Domaszewicz, “Tree-based anycast for wireless sensor/actuator networks,” in Proceedings of the 9th International Conference on Distributed Computing and Networking, Lecture Notes in Computer Science, Kolkata, India, 2008, SECTION: Sensor networks.
[19] S. S. Satapathy and N. Sarma, “TREEPSI: tree based energy efficient protocol for sensor information,” in Proceedings of the IFIP International Conference on Wireless and Optical Communications Networks, April 2006.
[20] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, “On maximizing the lifetime of delay-sensitive wireless sensor networks with anycast,” in Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08), pp. 1481–1489, April 2008.
[21] X. Wang, J. Wang, K. Lu, and X. Xu, “GKAR: a novel geographic (K)-anycast routing for wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, pp. 916–925, 2013.
[22] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, “Minimizing delay and maximizing lifetime for wireless sensor networks with anycast,” IEEE/ACM Transactions on Networking, vol. 18, no. 2, pp. 515–528, 2010.
[23] N. T. Dinh and Y. Kim, “Directional anycast routing in wireless sensor and actor networks,” Proceedings of the International Symposium on Communications and Information Technologies (ISCIT ‘12), pp. 251–255, 2012.
[24] G. Lu, B. Krishnamachari, and C. Raghavendra, “An adaptive energy-efficient and low-latency mac for data gathering in sensor networks,” in Proceedings of the Workshop on Energy-Efficient Wireless Communications and Networks (EWCN ’04), held in conjunction with the IEEE International Performance Computing and Communications Conference (IPCCC ‘04), April 2004.
[25] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proceedings of the
[26] Y. Gu and T. He, “Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links,” in Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’07), pp. 321–334, November 2007.

[27] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle MAC with scheduled channel polling,” in Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys ’06), pp. 321–334, November 2006.

[28] Y. Gu, T. He, M. Lin, and J. Xu, “Spatiotemporal delay control for low-duty-cycle sensor networks,” in Proceedings of the Real-Time Systems Symposium (RTSS ’09), pp. 127–137, December 2009.

[29] D. Noh, I. Yoon, and H. Shin, “Low-latency geographic routing for asynchronous energy-harvesting WSNs,” Journal of Networks, vol. 3, no. 1, pp. 78–85, 2008.

[30] G. Sun and B. Xu, “Delay-aware routing in low duty-cycle wireless sensor networks,” in Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’10), September 2010.

[31] Y. Gu and T. He, “Bounding communication delay in energy harvesting sensor networks,” in Proceedings of the 30th IEEE International Conference on Distributed Computing System (ICDCS ’10), pp. 837–847, June 2010.

[32] Y. Gu, T. Zhu, and T. He, “ESC: energy synchronized communication in sustainable sensor networks,” in Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP ’09), pp. 52–62, October 2009.

[33] W. K. G. Seah, A. E. Zhi, and H.-P. Tan, “Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)— survey and challenges,” in Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless (VITAE ’09), pp. 1–5, May 2009.

[34] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: survey and implications,” IEEE Communications Surveys and Tutorials, vol. 13, no. 3, pp. 443–461, 2011.

[35] M. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmy, “Efficient geographic routing over lossy links in wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 4, no. 3, article 12, 2008.

[36] Z. A. Eu, H.-P. Tan, and W. K. G. Seah, “Opportunistic routing in wireless sensor networks powered by ambient energy harvesting,” Computer Networks, vol. 54, no. 17, pp. 2943–2966, 2010.
