Collection tree-oriented mesh routing optimization for extending the lifetime of wireless sensor networks

Li-Yong Yuan¹ and Fei-Long Lin²

Abstract
Routing optimization in wireless sensor networks facilitates to reduce the overhead of the maintaining of wireless sensor networks and extend the lifetime of wireless sensor networks. Collection tree-based routing protocol, which does not require route discovery, has been widely used for low overheads of calculation and storage. However, with collection tree-based routing protocol, some nodes easily become the bottleneck points and quickly run out of the energy. To deal with this drawback, this article proposes a collection tree-oriented mesh routing strategy with cooperatively consuming the residual energy among the neighboring sensor nodes. The collection tree-oriented mesh routing is formulated into a linear programming problem with the purpose to maximize the network lifetime. By solving the optimization problem, the optimal mesh routing and data forwarding scheme is derived. Experimental simulations show that the proposed collection tree-oriented mesh routing optimization strategy can extend the network lifetime by more than 20%.

Keywords
Wireless sensor networks, network lifetime, IEEE 802.15.5, mesh cooperation, routing protocol

Date received: 13 April 2021; accepted: 3 February 2022

Handling Editor: Miguel Acevedo

Introduction

With the advantages in terms of low power consumption, low cost, fast deployment, and self-organization, wireless sensor networks (WSNs) have been widely used in many fields such as automation, environmental monitoring, and healthcare. A typical WSN consists of a certain number of sensor nodes, and each node is generally equipped with a data sensing module, data processing module, wireless communication module, and power module. Accordingly, WSNs still face the challenges such as fault tolerance, security, load balancing, energy consumption, and network lifetime. Among them, energy consumption and network lifetime have received particular attentions in the recent literature, since sensor nodes are generally powered by batteries with limited energy. The network lifetime, as defined in Liang and colleagues, depends on the minimum lifetime of the node in a WSN.

To extend the network lifetime, collection tree-based routing is a widely accepted solution which features low computation, low storage, and no route discovery requirement. Typically, in collection tree-based routing, each node forwards the sensed data and the data received from its child nodes to its parent node, that is, the data eventually reach the root node along the collection tree. However, collection tree-based routing still has two obvious drawbacks: (1) it is not reliable since the failure of a node will prevent the nodes in its sub-tree from transmitting data to the gateway node.
and (2) the network lifetime of a collection tree-based network is limited by bottleneck nodes whose energy is exhausted earliest because of less energy or larger traffic load, and the residual energy of the neighbor nodes of the bottleneck nodes cannot be utilized by collection tree routing.

In order to solve the reliability problem of collection tree-based routing, a mesh routing standard IEEE 802.15.5\(^7\) suitable for WSNs has been proposed. It allows each node to maintain the link information of its two-hop neighborhood in addition to the collection tree topology information, thus forming a mesh network with a collection tree as its backbone. The mesh routing strategies of IEEE 802.15.5 can be described as follows: (1) if the destination node is a one-hop neighbor, the node will forward the data packet to the destination node directly; (2) if the destination node is a two-hop neighbor of the current node, it will forward the data packet to the common neighbor between the destination node and the current node; and (3) otherwise, the node will select a node closest to the destination node through tree routing from its two-hop neighbors as an anchor node, and then forward the data packet to the common neighbor between the current node and the anchor node. However, the mesh routing by IEEE 802.15.5 does not consider how to optimize the traffic load of nodes to extend the network lifetime, and this problem has been seldom studied yet.

In this article, a collection tree-oriented mesh routing strategy is proposed and the network lifetime maximization problem based on collection tree-oriented mesh routing is modeled as a linear programming problem. The optimal mesh routing data forwarding scheme of each node is obtained by solving the linear programming problem. The effectiveness of the proposed method is verified by extensive simulation experiments.

The innovation and contribution of this article are summarized as follows:

1. A collection tree-oriented anchor node selection and mesh routing strategy is proposed to overcome the defects of IEEE 802.15.5-based mesh routing.
2. The collection tree-oriented mesh routing is formulated into a linear programming problem with the purpose to maximize the network lifetime. The optimal mesh routing and data forwarding scheme is derived by solving the optimization problem.

The remainder of this article is organized as follows. The review of the related works is presented in section “Related works,” the collection tree-oriented mesh routing optimization strategy is introduced in details in section “Collection tree-oriented mesh routing optimization strategy,” the performance evaluation is presented in section “Experimental simulation and result analysis,” and this article is finally concluded in section “Conclusion.”

### Related works

Sensor nodes are generally powered by batteries with very limited energy. It is difficult and even impossible to replace or recharge batteries for a large number of sensor nodes deployed in hazardous areas or in the wild. Therefore, optimizing network lifetime has attracted many researchers.\(^8\) The energy consumption of wireless communication is the main energy consumption of sensor nodes. Balancing the energy consumption between nodes by optimizing the data routes can effectively improve the network lifetime.\(^9\)

Collection tree routing is widely used in WSNs, especially in data collecting applications of WSNs.\(^10\) A lot of research works have been done on optimizing the collection tree for maximizing network lifetime. The literature\(^11–13\) studied the lifetime maximization of WSNs with data aggregation at relay nodes. The literature\(^11\) defined the network lifetime as the lifetime of the earliest dead node, proving that constructing the maximum network lifetime collection tree is a nondeterministic polynomial (NP) problem, and proposing a method which iteratively adjusts any given collection tree to reduce the energy consumption of the bottleneck nodes. The literature\(^12\) studied the problem of constructing the shortest path tree with the largest network lifetime in WSNs, and proposed a method selecting the maximum lifetime tree from the shortest path trees. An approximation algorithm for constructing the maximum lifetime data aggregation tree is proposed in the literature.\(^13\) The literature\(^5\) proved that constructing the maximum network lifetime collection tree in WSNs without data aggregation is an NP-hard problem, and proposed a heuristic collection tree construction method named Maximum network lifetime (MNL). In Liang et al.,\(^6\) constructing the maximum lifetime collection tree is expressed as the problem of minimizing the maximum weight of a spanning tree, an iterative algorithm named MITT(Maximum Lifetime Tree construction for data gaThering without aggregation) is proposed. Given a collection tree, MITT iteratively correlates the descendant nodes of the bottleneck nodes to their neighbor nodes with smaller energy consumption. The simulation results show that MITT is better than the heuristic construction methods such as Power efficient data gathering and aggregation protocol-Power aware (PEDAPPA)\(^14\) and MNL. A collection tree construction algorithm for maximizing the network lifetime named RaSMaLai is proposed in Imon et al.\(^15\) RaSMaLai also iteratively adjusts a given collection tree to optimize the network lifetime. The literature\(^16\) models constructing the maximum network...
lifetime collection tree as a mixed integer linear programming (MIP) problem, and converts the MIP to a linear programming problem by transforming each nonlinear equation in the MIP into a set of linear equations. In Liu and Chen, a two-stage approach is proposed to construct collection tree that works well under all worst-case scenarios for energy-harvesting WSNs. The literature proves that constructing the maximum network lifetime collection tree is NP-complete by reducing the set cover problem to it and proposes an algorithm named Balanced energy consumption Data Collection Tree (BDCT) to solve it.

However, no matter how the collection tree is optimized, it has two major problems as mentioned in section “Introduction.” In order to solve the reliability problem of the collection tree routing, IEEE 802.15.5, a mesh standard suitable to WSNs, is proposed. IEEE 802.15.5 allows each node to maintain the link information of several hop neighbors in addition to the information of the collection tree topology, thus forming a mesh network with the collection tree as its backbone. As described in section “Introduction,” the mesh routing strategy of IEEE 802.15.5 preserves the advantages of collection tree routing and improves the reliability of networks. However, the IEEE 802.15.5 standard focuses on improving the reliability of networks, and does not consider how to balance traffic load between nodes through mesh routing. From the perspective of extending the network lifetime, it mainly has two problems: (1) IEEE 802.15.5 mesh routing always selects a two-hop neighbor node closest to the destination node along collection tree routing as the anchor node. This anchor node selection strategy may cause a decrease of network lifetime. For example, in the network shown in Figure 1, where the larger nodes have more energy, the black arrow lines indicate the collection tree routes, and the dotted lines indicate the adjacency relationships between nodes. According to the anchor node selection strategy of IEEE 802.15.5, node \( H \) will select gateway node \( v_0 \) as the anchor node, thus selecting node \( D \) as the relay node, that is, the data packages originating from node \( H \) will flow along the path shown by the red curves to the gateway node, which increases the traffic load of node \( D \) and may reduce the network lifetime. (2) The mesh routing of IEEE 802.15.5 randomly selects a node as the relay node without considering the traffic load and energy consumption balance of the multiple common neighbors between the anchor node and the current node.

The problem of balancing traffic load and energy consumption has been widely studied to prolong the lifetime of WSNs. A cluster-based balanced energy consumption algorithm (BECA) is proposed in Qin et al. BECA introduces in multiple inter-cluster links to distribute the load. In order to minimize and balance the energy consumption, a grid-based reliable multi-hop routing approach for WSNs is proposed in Chen and Shen. The literature proposes a lightweight load-balancing and verification scheme (secure load and energy balancing) based on clustered WSNs. For the data collection scenario of large-scale WSNs with multi-gateway nodes, the literature proposes a reactive and adaptive load-balancing (RALB) algorithm to balance the traffic load of gateways while balancing the in-network traffic load. A collaborative load-balancing algorithm (CoLBA) is proposed in Tall et al., CoLBA is a queuing delay-based routing protocol that avoids packet queue overflow and uses a prediction approach to optimize control messages transmission. A novel load-balancing strategy for data transmission of WSNs is proposed in Liu and Zhang, which makes the full use of the advantages of super nodes with more powerful hardware and greater communication capacity to realize data traffic redistribution. The literature presents a new multi-chain routing strategy named the destination-oriented routing algorithm (DORA), which generates a new multi-chain routing scheme to transmit packets for energy-balanced WSNs.

Unfortunately, there is no literature addressing the above two issues in mesh routing of IEEE 802.15.5. In order to extend the lifetime of wireless sensor mesh networks, this article proposes a collection tree-oriented mesh routing strategy, and optimizes the data traffic of outgoing links for each node according to their energy.

**Collection tree-oriented mesh routing optimization strategy**

**Network model and overall scheme**

In this article, the network is assumed as follows: (1) once the sensor nodes are deployed, their position will
not change; (2) there is only one gateway node; (3) the gateway node has sufficient energy, and the other nodes are powered by batteries with limited energy; and (4) there is at least one path between the gateway node and the other nodes, that is, there are no isolated nodes in the network. The WSN can be represented as an undirected graph $G(V, A)$, where $V$ is a set of nodes, including a gateway node denoted by $v_0$, and $A$ represents a set of links, that is, $A = \{ <u, v> | d(u, v) \leq d_0, u, v \in V \}$, where $d(u, v)$ represents the link and the distance between nodes $u$ and $v$, respectively, and $d_0$ represents the communication radius of nodes. $N(u)$ represents all one-hop neighbors of node $u$, that is, $N(u) = \{ v | d(u, v) \leq d_0, v \in V \}$. $T(V_T, A_T)$ represents a spanning tree rooted at $v_0$ in $G(V, A)$, where $V_T$ and $A_T$ represent the set of nodes and the set of links in the spanning tree $T$, respectively. $p(v)$ represents the parent node of node $v$. $TL(v)$ represents the tree layer of node $v$ in tree $T$, which is equal to the tree layer of its parent node plus 1, that is, $TL(v) = TL(p(v)) + 1$, and $TL(v_0) = 0$. In addition, $AN_v$ represents all ancestor nodes of node $v$, which includes the parent node of node $v$. $DN_v$ represents all descendant nodes of node $v$, which includes the children of node $v$.

Similar to IEEE 802.15.5 mesh routing, the collection tree-oriented mesh routing strategy proposed in this article also selects one node among the two-hop neighbors of the current node as the anchor node. Then, by transmitting data to multiple relay nodes between the current node and the anchor node, some data are allowed to bypass the bottleneck nodes in the collection tree, thus reducing the traffic load on the bottleneck nodes and prolonging the network lifetime. The overall scheme for extending the network lifetime based on collection tree-oriented mesh routing is as follows: first, a collection tree with the maximum network lifetime is constructed using BDCT proposed in John et al.; then, the data forwarding scheme of each node is optimized by exploiting the method described in section “Mesh routing optimization for extending the network lifetime” of section “Collection tree-oriented mesh routing optimization strategy”; finally, the nodes in the network forward data according to the optimized data forwarding scheme.

### Collection tree-oriented anchor node selection and mesh routing strategy

As described in section “Related works,” the anchor node selection strategy of IEEE 802.15.5 has the potential to make mesh routing-based network lifetime lower than collection tree routing. To avoid this problem, we propose a collection tree-oriented anchor node selection and mesh routing strategy: first, find the minimum tree layer ancestor node of the data source node from the one-hop neighbors of the current node, and select the parent node of the minimum tree layer ancestor node as the anchor node. Then, the data packages are forwarded to a common neighbor between the current node and the anchor node according to an optimized data forwarding scheme.

We give an example shown in Figure 2 to further introduce the collection tree-oriented anchor node selection and mesh routing strategy. In Figure 2, the larger nodes have more energy, and the dotted lines indicate the adjacency relationships between nodes, and the black arrow lines indicate the collection tree routes, and the dotted lines with arrows indicate collection tree-oriented mesh routes, and node $F$ is the bottleneck node in the collection tree, and node $N$ wants to transmit the sensed data packages to the gateway node $v_0$. According to the collection tree-oriented anchor node selection strategy, node $N$ selects node $I$ as the anchor node, since node $H$ is its only ancestor node in its one-hop neighbors and node $I$ is the parent node of node $H$. Nodes $H$ and $L$ are the common neighbors of node $N$ and the anchor node $I$, so node $N$ forwards a portion of data packages to node $H$, and forward another portion of data packages to node $L$. After receiving the data packages originating from node $N$, node $H$ selects node $F$ as the anchor node, because node $I$ is the only ancestor node of node $N$ in node $H$’s one-hop neighbors and node $F$ is the parent node of node $I$. Since node $I$ is the only common neighbor of node $H$ and the anchor node $F$, node $H$ forwards all data packages originating from node $N$ to node $I$. After receiving the data packages originating from node $N$, node $L$ also selects node $F$ as the anchor node, because node $H$ and node $I$ are both ancestor nodes of node $N$ in the one-hop neighbors of

---

**Figure 2.** An example of the collection tree-oriented anchor node selection and mesh routing strategy.
node $L$ and the tree layer of node $I$ is smaller than node $H$, so the parent node of node $I$, that is, node $F$, is selected as the anchor node. Since node $I$ and node $J$ are the common neighbors of node $L$ and the anchor node $F$, node $L$ forwards a portion of data packages to node $I$ and forwards another portion of data packages to node $J$. After receiving the data packages originating from node $N$, node $I$ selects the gateway node $v_0$ as the anchor node, since node $F$ is the earliest ancestor node of node $N$ in node $I$’s one-hop neighbors, and the parent node of node $F$ is node $v_0$. Node $E$ and node $F$ are the common neighbors of node $I$ and node $v_0$, so node $L$ forwards a portion of data packages to node $E$ and another portion of data packages to node $F$, thus reducing the traffic load of node $F$ and extending the network lifetime. After receiving the data packages originating from node $N$, node $J$ selects node $v_0$ as the anchor node, because node $F$ is the earliest ancestor node of node $N$ in the one-hop neighbors of node $J$, and the parent node of node $F$ is node $v_0$. Since node $F$ is the only common neighbor of node $J$ and node $v_0$, node $J$ will forward all data packages to node $F$. Nodes $E$ and $F$ are the one-hop neighbors of node $v_0$, so they directly send data packages to node $v_0$.

As shown in the above example, the collection tree-oriented anchor node selection and mesh routing policies always give data packets a chance to reach gateway nodes along the collection tree. In the worst case, the network lifetime of the collection tree-based mesh routing is no less than that of collection tree routing.

**Mesh routing optimization for extending the network lifetime**

As described in section “Collection tree-oriented anchor node selection and mesh routing strategy” of section “Collection tree-oriented mesh routing optimization strategy,” when there are multiple common neighbors (called relay nodes later) between the current node and the anchor node, the current node forwards different numbers of data packets to different relay nodes. Obviously, different data forwarding scheme will result in different traffic load of the relay nodes, which mean different lifetime of nodes. The collection tree-oriented mesh routing optimization strategy is to optimize the traffic on links from nodes to their relay nodes with the goal of maximizing the lifetime of the minimum lifetime node.

$CN(u, v)$ represents all the common neighbors between node $u$ and node $v$, $\kappa(u, x, v)$ represents the data rate at which node $u$ forwards data to node $x \in CN(u, v)$, where node $v$ is an anchor node, which is a variable to be optimized (referred to as traffic variable). According to the collection tree-oriented anchor node selection and mesh routing strategy, there are two types of data packages that need nodes to transmit: one is the data packages generated by them, and the other is the data packages that need them to relay. For the data packages generated by node $u$, the anchor node is its grandfather node, denoted by $g(u)$. For any data package in traffic $\kappa(x, u, w)$ flowing into node $u$, node $w$ is the ancestor node of the source nodes of the data package, it is also the one-hop neighbor of node $u$. According to the collection tree-oriented anchor node selection and mesh routing strategy, the anchor node for traffic $\kappa(x, u, w)$ is $p(\arg_{y \in [CN(u, v_0)]} \min TL(y))$. Therefore, all the anchor nodes of node $u$ are $\Lambda(u) = \{p(\arg_{y \in [CN(u, v_0)]} \min TL(y))\} \cup \{g(u)\}$.

The meaning of $\Lambda(.)$ is further explained by the example shown in Figure 2. For node $H$, in addition to forwarding the data generated by itself, it also needs to forward the traffics $\kappa(N, H, I)$, $\kappa(K, H, C)$, and $\kappa(G, H, D)$ which flow into node $H$. For forwarding $\kappa(N, H, I)$, node $H$ selects node $F$ as the anchor node, for node $I$ is the only ancestor node of the data source node in the one-hop neighbors of node $H$ and the parent node of node $I$ is node $F$. For forwarding $\kappa(K, H, C)$, node $H$ selects node $v_0$ as the anchor node, because nodes $C$ and $D$ are the ancestor nodes of the data source node, and the tree layer of node $D$ is the smaller than node $C$, and the parent node of node $D$ is $v_0$. Similarly, for forwarding $\kappa(G, H, D)$, node $H$ selects node $v_0$ as the anchor node. So, $\Lambda(H) = \{F, v_0\}$.

$\kappa(u)$ represents all the traffic variables flowing out from node $u$, $\min TL(u) > 1$, that is, $\kappa(u) = \{\kappa(u, x, v) | x \in CN(u, v), \min TL(u) > 1\}$, $\kappa(u)$ denotes all traffic variables flowing out from nodes with tree layer greater than 1, that is, $\kappa = \sum_{\kappa(u, x, v) \in \kappa(u, x, v)} \kappa(u, x, v)$. The neighbor nodes of node $v_0$ do not need other nodes to relay data. In particular, we use the set $\Phi = \{\phi(x, v_0) | x \in N(v_0)\}$ to represent all the traffic variables flowing out from nodes with tree layer equal to 1.

This article focuses on the data collection application of WSNs without data aggregation, where each node generates data at the same rate $\sigma$. For node $v$ with tree layer equal to 1, the total outflow data rate is $\phi(v, v_0)$, and the total inflow data rate is $\sum_{u \in V} \kappa(u, v, v_0)$. For node $v$, the total outflow data rate should be equal to the total inflow data rate plus $\sigma$. Therefore, the following traffic constraints should be satisfied

$$\phi(v, v_0) = \sum_{u \in V} \kappa(u, v, v_0) + \sigma \quad \forall v \in V_T, TL(v) = 1 \quad (1)$$

For a node $v$ whose tree layer is greater than 1, the total outgoing traffic with node $v$ as the anchor node can be expressed as $\sum_{x \in CN(v, y)} \kappa(v, x, y)$. According to the
collection tree-oriented anchor node selection strategy, node \( v \) always selects node \( y \) as the anchor node for any data traffic originating from the descendant nodes of node \( y \) and flowing into node \( v \), that is

\[
\sum_{x \in CN(v, y)} \kappa(v, x, y) = \sigma \times b_{v, y} + \sum_{w \in DN_y} \kappa(u, v, w)
\]

\[\forall y \in \Lambda(v), \forall v \in V_T, TL(v) > 1\]  

(2)

where

\[
b_{v, y} = \begin{cases} 
1 & \text{when } g(v) = y \\
0 & \text{otherwise}
\end{cases}
\]

which indicates that the outgoing traffic destined to \( y \) should contain the data generated by node \( v \) when node \( y \) is the grandfather of node \( v \).

The meaning of formula (2) is further explained by the example shown in Figure 2. For node \( H \), \( \Lambda(H) = \{F, v_0\} \). Node \( H \) will select node \( v_0 \) as the anchor node for \( \kappa(K, H, C) \) and \( \kappa(G, H, D) \), and the traffic flowing out from node \( H \) with node \( v_0 \) as the anchor node is \( \kappa(H, D, v_0) \). Therefore, \( \kappa(K, H, C) + \kappa(G, H, D) = \kappa(H, D, v_0) \). Node \( H \) will select node \( v_0 \) as the anchor node for \( \kappa(N, H, I) \), and the traffic flowing out from node \( H \) with node \( F \) as the anchor node is \( \kappa(H, I, F) \). Since node \( F \) is the grandfather of node \( H \), the traffic \( \kappa(H, I, F) \) should contain the data generated by node \( H \). Therefore, \( \kappa(N, H, I) + \sigma = \kappa(H, I, F) \). Table 1 lists the traffic constraints for all nodes in the network shown in Figure 2.

\( E_r \) represents the initial energy of node \( v \), and \( e_0 \) represents the energy consumption for receiving one-bit data, and \( e_1 \) represents the energy consumption for transmitting one-bit data. The total inflow data rate of node \( v \) is \( \sum_{u \in V \setminus \{v\}} \kappa(u, v, w) \). The total outgoing data rate of node \( v \) is \( \sum_{u \in V \setminus \{v\}} \kappa(u, v, w) \). The energy consumption rate of node \( v \) for receiving and transmitting data is \( e_1 \times \sigma + \sum_{u \in V \setminus \{v\}} \kappa(u, v, w) \times (e_0 + e_1) \). Similar to the literature, this article only considers the energy consumption of data transmission and reception. Therefore, the lifetime of node \( v \) can be expressed as

\[
L_v = \frac{E_v}{e_1 \times \sigma + \sum_{u \in V \setminus \{v\}} \kappa(u, v, w) \times (e_0 + e_1)}
\]

(3)

As mentioned above, this article defines the network lifetime \( L_{net} \) as the lifetime of the earliest dead node, that is

\[
L_{net} = \min_{v \in V \setminus \{v_0\}} L_v
\]

(4)

Therefore, the network lifetime maximization problem based on collection tree-oriented mesh routing can be formulated as the following optimization problem

\[
\text{maximize } L_{net} \quad \text{w.r.t. : } \Phi, \kappa
\]

s.t. : (1), (2), (3), (4)

\[
\kappa(u, v, w) \geq 0, \forall (u, v, w) \in \kappa
\]

\[
\phi(v, v_0) \geq 0, \forall \phi(v, v_0) \in \Phi
\]

(6)

From equalities (3) and (4), we can get

\[
L_{net} \leq \sum_{u \in V \setminus \{v\}} \kappa(u, v, w) \times (e_0 + e_1) + e_1 \times \sigma
\]

(7)

Table 1. An example of traffic constraints.

| Node | Anchor node | Traffic constraint |
|------|-------------|--------------------|
| A    | N/A         | \( \kappa(C, A, v_0) + \kappa(b, A, v_0) + \sigma = \phi(A, v_0) \) |
| B    | \( y_0 \)   | \( \kappa(G, C, D) + \sigma = \kappa(C, A, v_0) + \kappa(C, D, v_0) \) |
| C    | \( v_0 \)   | \( \kappa(C, D, v_0) + \kappa(H, D, v_0) + \sigma = \phi(D, v_0) \) |
| D    | N/A         | \( \kappa(l, E, v_0) + \sigma = \phi(E, v_0) \) |
| E    | N/A         | \( \kappa(l, F, v_0) + \kappa(l, j, F, v_0) + \sigma = \phi(f, v_0) \) |
| F    | N/A         | \( \kappa(K, G, C) + \sigma = \kappa(G, H, D) + \kappa(G, C, D) \) |
| G    | D           | \( \kappa(N, H, I) + \sigma = \kappa(H, I, F) \) |
| H    | F           | \( \kappa(K, H, C) + \kappa(G, H, D) = \kappa(H, D, v_0) \) |
| I    | \( y_0 \)   | \( \kappa(H, I, F) + \kappa(L, I, F) + \sigma = \kappa(l, E, v_0) + \kappa(l, F, v_0) \) |
| J    | \( v_0 \)   | \( \kappa(L, j, F) + \kappa(O, l, I) + \sigma = \kappa(j, f, v_0) \) |
| K    | C           | \( \kappa(M, K, G) + \sigma = \kappa(K, G, C) + \kappa(K, H, C) \) |
| M    | G           | \( \sigma = \kappa(M, K, G) \) |
| N    | \( I \)     | \( \sigma = \kappa(N, H, I) + \kappa(N, L, I) \) |
| O    | \( I \)     | \( \sigma = \kappa(O, L, I) + \kappa(O, J, I) \) |

N/A: not available.
Let $\tau = 1/L_{net}$, then inequality (7) can be transformed into

$$\sum_{\kappa(u, v, w) \in \kappa} k(u, v, w) \times (e_0 + e_1) + e_1 \times \sigma \leq \tau E_v$$

(8)

The above optimization problem can be transformed into the following form

minimize $\tau$

w.r.t.: $\Phi, \kappa$

s.t.: (1), (2), (5), (6), (8)

Inequalities (1) and (2) represent the traffic constraints, and inequalities (5) and (6) indicate that the traffic variables cannot be less than 0, and inequality (8) represents the energy constraint of nodes. The above optimization problem is a linear programming problem that can be solved by existing tools such as the `linprog` function in MATLAB. Suppose $\kappa^* = \{\kappa^*(u, x, v) | \forall u \in V_T, TL(u) > 1\}$ is the optimal solution to the above optimization problem. For an anchor node $v$, the optimal probability $(u, x, v)$ of node $u$ forwarding data to the relay node $x$ can be expressed as

$$(u, x, v) = \frac{\kappa^*(u, x, v)}{\sum_{v \in CN(u, v)} \kappa^*(u, y, v)}$$

(9)

$\forall v \in \Lambda(u), \forall u \in V_T, TL(u) > 1$

**Implementation scheme**

When deploying a WSN, a distributed collection tree construction method (such as the collection tree construction method of IEEE 802.15.5) is used to construct a collection tree. Each node collects information of one-hop neighbors and sends the one-hop neighbors’ information and node energy to the gateway node through the collection tree routes. The gateway node constructs a maximum network lifetime collection tree using BDCT according to the collected information. Then, the optimization method proposed in section “Mesh routing optimization for extending the network lifetime” is used to solve the optimal forwarding scheme of each node. The gateway node sends the collection tree topology information and the optimized forwarding scheme to each node.

In the network initialization stage described above, the energy consumption of nodes except the gateway node is approximately equivalent to two rounds of data collection. More complex calculations, such as building a collection tree and solving the network lifetime optimization problem, are performed at gateway node with sufficient energy, storage space, and computing power.

During the data collection stage, the nodes make routing decisions based on the received forwarding schemes. The routing decision process of nodes when forwarding data is given as follows:

Step 1: if the current node $u$ is a one-hop neighbor of the gateway node, node $u$ directly forwards the data packet to the gateway node; otherwise, go to Step 2.

Step 2: the current node $u$ uses the following algorithm to determine the anchor node $v$ for forwarding the data packet originating from node $s$.

Step 3: the current node $u$ selects a node $x \in CN(u, v)$ as the relay node according to the data forwarding probabilities $\{\phi(u, x, v) | x \in CN(u, v)\}$, and forwards the data packet to the relay node.

To save the collection tree topology information and the optimized forwarding scheme, each node maintains a route table besides an extended neighbor list (EN-list) and a connectivity matrix for all nodes in EN-list defined in IEEE 802.15.5.\(^7\) An example of the route table is shown in Table 2.

**Experimental simulation and result analysis**

In this article, we focus on traffic optimization based on collection tree-oriented mesh routing without considering communication interference, for communication interference can be avoided by a reasonable sleep scheduling strategy which is beyond the scope of this article. Based on the above considerations, we use Java programming language and MATLAB to make the simulation programs. A large number of simulations have been done to verify the effectiveness of the proposed method. The effects of the number of nodes, node density, and energy difference of nodes on the performance of the proposed method are evaluated by simulation experiments, where node density is defined as the number of nodes in an area of $\pi d^2$, and energy difference of nodes is defined as the ratio of the maximum energy of nodes to the minimum energy of nodes. For each network setting in which the number of nodes, node densities, and energy difference of nodes are determined, 100 networks are randomly generated. For each network, BDCT and the collection tree-oriented mesh routing optimization method proposed in this article (referred
Table 2. An example of the route table.

| Anchor node | Relay node | Forwarding probability |
|-------------|------------|------------------------|
| \( v_0 \)   | A          | 0.4                    |
| \( v_0 \)   | D          | 0.6                    |
| ...         | ...        | ...                   |

Table 3. The parameter settings.

| Parameters | Values       |
|-----------|--------------|
| \( e_0 \) | 50 nJ/bit\(^{13}\) |
| \( e_1 \) | 100 nJ/bit\(^{13}\) |
| \( S \)   | 0.1 bps     |

Figure 3. An example of collection tree routes and mesh routes: (a) collection tree routes and (b) mesh routes.

Figure 4. The performance comparison of CToMR to BDCT.

energy, and the largest node represents the gateway node, and the dotted lines represent the adjacency relationships, and the red solid lines represent the data traffic, and the thickness of lines indicates the size of traffic, and the node marked with an ellipse in Figure 3(a) represents the bottleneck node of the collection tree.

Figure 4 shows the comparison of CToMR to BDCT in 50 simulations, where node density is 10, the number of nodes is 100, and the energy difference of nodes is 5. Figure 4 shows that the network lifetime based on CToMR is no less than BDCT in each simulation. In fact, this phenomenon was observed in all the simulations we have done. In other words, in the worst case, the network lifetime of the collection tree-based mesh routing is not lower than the collection tree routing. The reason behind this phenomenon is that the data forwarding scheme based on collection tree is one of the solutions which meet the traffic and energy constraints of the network lifetime optimization problem and the data forwarding scheme obtained by CToMR is the optimal one of all solutions.

Figure 5 shows the performance comparison of CToMR to BDCT in simulations where the node density is 10 and the energy difference of nodes is 5. The results shown below are the average of 100 randomly
generated networks. The “.” and “+” in the following figures, respectively, indicate the lower bound and the upper bound of the confidence interval with a confidence level of 0.95. As shown in Figure 5, the performance of CToMR decreases as the network size increases. This is because as the network size increases, the number of bottleneck nodes increases, and the probability that at least one bottleneck node cannot get help from its neighbors will increase. As a result, the performance of CToMR is degraded.

Figure 6 shows the performance comparison of CToMR to BDCT under different node densities. It can be seen from Figure 6 that the performance of CToMR increases as the node density increases, and the reason is that the probability that the bottleneck nodes get help from their neighbors increases as the number of neighbor nodes increases.

Figure 7 shows the performance comparison of CToMR to BDCT under different energy difference of nodes. The performance of CToMR increases as the energy difference of nodes increases. This is because the minimum energy of nodes has a greater impact on the lifetime of collection tree network generated by BDCT, and the minimum energy of nodes has a smaller impact on the network lifetime based on CToMR since CToMR can reduce
the traffic load of bottleneck nodes by letting some data packages bypass the bottleneck nodes through mesh routing. Therefore, the performance of CToMR relative to BDCT increases as the energy difference of nodes increases.

Figures 5–7 show that CToMR can extend the network lifetime by more than 22% compared with BDCT. The cost of CToMR is that each node needs to maintain its one-hop neighbor information and some data forwarding information. This article adopts the block address allocation strategy of IEEE 802.15.5, that is, the information of each one-hop neighbor contains 4 bytes of block address, 1 byte of tree layer, and 2 bytes of address of its parent. A data forwarding information requires 5 bytes, including 2 bytes of relay node address, 2 bytes of anchor node address, and 1 byte of forwarding probability. Figures 8–10 show the average memory usage of nodes under different number of nodes, difference of nodes, and node density, respectively.

As shown in Figures 8–10, the number of nodes and the energy difference of the nodes have a smaller impact on the average memory usage of nodes, and the node density has a larger impact. As the node density increases, each node needs to maintain more one-hop neighbor information and more data forwarding information because the number of common nodes between nodes and their anchor nodes increases. For sensor nodes with tens of Kbyte or even hundreds of Kbyte of memory, the storage overhead required by CToMR is completely affordable.

**Conclusion**

This article proposes a mesh routing strategy based on collection tree routing for optimization of network lifetime of WSNs. It allows some traffic the opportunity to bypass bottleneck nodes in the collection tree, and ensures that data always have a chance to reach the gateway node along the collection tree. That is, in the worst case, the network lifetime based on the collection tree-oriented mesh routing is not lower than the collection tree routing. By modeling the network lifetime maximization problem based on the collection tree-oriented mesh routing as a linear programming problem, the optimal mesh routing data forwarding scheme of each node is obtained by solving the linear programming problem. Simulation results show that, compared to the collection tree routing, the collection tree-oriented mesh routing optimization strategy can achieve more than 20% network lifetime extension with less storage cost.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported in part by the Zhejiang provincial Top Discipline of Cyber Security at Zhejiang Normal University.
University; in part by the NSF of China under grant 61877055; and in part by the NSF of Zhejiang Province under grant LY22F030006.

ORCID iD
Li-Yong Yuan https://orcid.org/0000-0002-6251-9866

References
1. Dong S, Sarem M and Zhou W. Distributed data gathering algorithm based on spanning tree. IEEE Syst J 2021; 15(1): 289–296.
2. Lin F, Dai W, Li W, et al. A framework of priority-aware packet transmission scheduling in cluster-based industrial wireless sensor networks. IEEE T Ind Inform 2020; 16(8): 5596–5606.
3. Malik NN, Alosaimi W, Uddin MI, et al. Wireless sensor network applications in healthcare and precision agriculture. J Healthc Eng 2020; 2020(1): 8836613.
4. Zhang Y, Zhu Y, Yan F, et al. Energy-efficient radio resource allocation in software-defined wireless sensor networks. IET Commun 2018; 12(3): 349–358.
5. Liang W and Liu Y. Online data gathering for maximizing network lifetime in sensor networks. IEEE T Mobile Comput 2007; 6(1): 2–11.
6. Liang J, Wang J, Cao J, et al. An efficient algorithm for constructing maximum lifetime tree for data gathering without aggregation in wireless sensor networks. In: Proceedings of the IEEE international conference on computer communications, San Diego, CA, 15–19 March 2010, pp.506–510. New York: IEEE.
7. Lee M, Zhang R, Zhu C, et al. Meshing wireless personal area networks: introducing IEEE 802.15.5. IEEE Commun Mag 2010; 48(1): 54–61.
8. Ma X, Dong H, Liu X, et al. An optimal communications protocol for maximizing lifetime of railway infrastructure wireless monitoring network. IEEE T Ind Inform 2018; 14(8): 3347–3357.
9. Mehrjoo S and Khunjush F. Optimal data aggregation tree in wireless sensor networks based on improved river formation dynamics. Comput Intell 2018; 34(3): 802–820.
10. Aparicio S, Martinez-Garrido MI, Ranz J, et al. Routing topologies of wireless sensor networks for health monitoring of a cultural heritage site. Sensors 2016; 16(10): 1732–1745.
11. Wu Y, Mao ZJ and Fahmy S. Constructing maximum-lifetime data-gathering forests in sensor networks. IEEE ACM T Network 2010; 18(5): 1571–1584.
12. Shan MF, Chen GH, Luo DJ, et al. Building maximum lifetime shortest path data aggregation trees in wireless sensor networks. ACM T Sensor Network 2014; 11(1): 11.
13. Lin HC and Chen WY. An approximation algorithm for the maximum-lifetime data aggregation tree problem in wireless sensor networks. IEEE T Wirel Commun 2017; 16(6): 3787–3798.
14. Tan HÓ and Körpeoiülu I. Power efficient data gathering and aggregation in wireless sensor networks. SIGMOD Rec 2003; 32(4): 66–71.
15. Imon SKA, Khan AH, Di Francesco M, et al. Energy-efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks. IEEE ACM T Network 2015; 23(5): 1401–1415.
16. Zhou F, Chen Z, Guo S, et al. Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs. In: Proceedings of the IEEE international conference on global communications, San Diego, CA, 6–10 December 2016, pp.1–6. New York: IEEE.
17. Liu RS and Chen YC. Robust data collection for energy-harvesting wireless sensor networks. Comput Netw 2020; 167: 107025.
18. John J, Kasbekar G and Baghini MS. Maximum lifetime convergecast tree in wireless sensor networks. Ad Hoc Netw 2021; 120: 102564.
19. Qin D, Ji P, Yang S, et al. An efficient data collection and load balance algorithm in wireless sensor networks. Wirel Netw 2019; 25(7): 3703–3714.
20. Chen Z and Shen H. A grid-based reliable multi-hop routing protocol for energy-efficient wireless sensor networks. Int J Distrib Sens N. Epub ahead of print 28 March 2018. DOI: 10.1177/1550147718765962.
21. Zhou J and Lin Z. Lightweight load-balanced and authentication scheme for a cluster-based wireless sensor network. Int J Distrib Sens N. Epub ahead of print 23 February 2021. DOI: 10.1177/1550147720980326.
22. Miao Y, Vural S, Sun Z, et al. A unified solution for gateway and in-network traffic load balancing in multi-hop data collection scenarios. IEEE Syst J 2016; 10(3): 1251–1262.
23. Tall H, Chalhoub G, Hakem N, et al. Load balancing routing with queue overflow prediction for WSNs. Wirel Netw 2019; 25(1): 229–239.
24. Liu X and Zhang P. Data drainage: a novel load balancing strategy for wireless sensor networks. IEEE Commun Lett 2018; 22(1): 125–128.
25. Wang K, Yu CM and Wang LC. DORA: a destination-oriented routing algorithm for energy-balanced wireless sensor networks. IEEE Internet Things 2021; 8(3): 2080–2081.