Wireless Netw (2016) 22:975–989
DOI 10.1007/s11276-015-1012-2

Distributed probabilistic routing for sensor network lifetime optimization

Yongcai Wang[1] · Haisheng Tan[2]

Published online: 9 July 2015

Abstract A probabilistic and distributed routing approach for multi-hop sensor network lifetime optimization is presented in this paper. In particular, each sensor self-adjusts their routing probabilities locally to their forwarders based on its neighborhood knowledge, while aiming at optimizing the overall network lifetime (defined as the elapsed time before the first node runs out of energy). The theoretical feasibility and a practical routing algorithm are presented. Specifically, a sufficient distributed condition regarding the neighborhood state for distributed probabilistic routing to achieve the optimal network lifetime is presented theoretically. Based on it, a distributed adaptive probabilistic routing (DAPR) algorithm, which considered both the transmission scheduling and the routing probability evolvement is developed. We prove quantitatively that DAPR could lead the routing probabilities of the distributed sensors to converge to an optimal state which optimizes the network lifetime. Further, when network dynamics happen, such as topology changes, DAPR can adjust the routing probabilities quickly to converge to a new state for optimizing the remained network lifetime. We presented the convergence speed of DAPR. Extensive simulations verified its convergence and near-optimal properties. The results also showed its quick adaptation to both the network topology and data rate dynamics.

✉ Haisheng Tan
hstan.hku@gmail.com

1 Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

2 Department of Computer Science, Jinan University, Guangzhou, China

Keywords Sensor networks · Network lifetime maximization · Probabilistic routing · Distributed algorithm

1 Introduction

Designing routing protocols to prolong network lifetime is an important problem in wireless sensor networks (WSN). Tremendous studies have been devoted to this problem. Centralized routing optimization approaches, which design the optimal routing strategy for sensors in offline phase [1, 2] suffer the overhead for centralized network information collection. Numerous distributed and practical routing protocols [3–5] have been proposed, in which sensors use link qualities to the neighbors, hop count, or left energy of the neighbors as metrics to optimize the routing selection. However, generally, the optimal network lifetime can hardly be achieved by distributed routing algorithms.

In this paper, we consider the sensor network lifetime optimization problem for the case when each sensor can online adjust its routing probabilities to its one-hop forwarders. From the centralized optimization point of view, the probabilistic routing scheme relaxes the routing problem from an integer programming to a linear programming problem. This is because the routing decision for each sensor is relaxed from determining a unique forwarder to optimizing routing probability assignments to the multiple forwarders[6]. Therefore, the optimal routing strategy for network lifetime optimization by probabilistic routing can be resolved centrally by a maximum flow approach [6].

However, we further consider the practical distributed routing scenario, when each sensor can only learn information from neighborhood and make routing decisions distributively. This raises an interesting question: "is it possible for the sensors to self-adjust their routing probabilities based on local information, so as to lead the network converge quickly to a state, where the remaining network lifetime is optimized?" We investigate this problem by a probabilistic routing model, in which each link $\langle i,j \rangle$ is assigned an adjustable probability $p_{i,j}(t)$, which indicates the probability of i routing data to j at time t. We propose a distributed routing probability updating algorithm which can converge quickly from arbitrary initial state to a stable state that can optimize the remaining network lifetime.

The practical value of this problem can be the selection of forwarding chances in multi-path or opportunistic routing [7] for network lifetime optimization. It is also of good value for the cases when packets can be divided, such as in network coding, so that the coded pieces can be transmitted to the forwarders following the calculated routing probabilities for prolonging the network lifetime. In this work, we mainly focus on the design and analysis of the distributed routing probability updating algorithm but not going into the details of its applications.

Firstly, we start from a general directed graph model for sensor network data collection. A new local cost function is carefully designed for each node so that it can be partially aware of faraway sensors' states via requiring neighbors' local costs. The cost function is a recursive function of a sensor's routing probabilities and its forwarders' local costs. Based on the function, a *distributed condition* indicating the states of routing probability assignments for remaining network lifetime optimization is derived. We show that when each sensor reaches the distributed condition, the sensors must have self-organized into some cost-equivalent sets, i.e., the nodes in each set have the same local cost.

Secondly, for network lifetime optimization, the distributed condition suggests a distributed routing probability updating problem to balance the local costs of sensors. Because every sensor adjusts routing probabilities independently without knowing other's concurrent operations, the local cost oscillation (or instability) and the problem of falling into local optimum have to be addressed. We present a virtually guided method to update the routing probability so that the oscillation and local optima can be avoided. Further, a distributed adaptive probabilistic routing algorithm (DAPR) is proposed, which considers not only the routing probability updating but also the transmission scheduling and information exchanging within the neighborhood.

Thirdly, we prove DAPR can guarantee the routing probabilities of all the sensors to converge to a stable state satisfying the distributed condition for network lifetime optimization. We further demonstrate DAPR can self-adapt to the network topology and flow dynamics and converge to a new optimal state quickly. The analysis of the convergence speed and speed-up schemes are also presented. Finally, extensive simulations are conducted to verify the stability, optimality and self-adapting characteristics of our algorithm.

The rest of the paper is organized as follows. The problem model is introduced in Sect. 2. The local cost function and the local conditions for network lifetime optimization are presented in Sect. 3. We propose the DAPR algorithm in Sect. 4. The properties of DAPR in static and dynamic networks are investigated in Sect. 5; Simulation results are in Sect. 6; Related works are presented in Sect. 7. We conclude the whole paper in Sect. 8.

2 Problem model

2.1 Network model

We model the sensor network as a general directed acyclic graph (DAG) $G = (V, E)$ for data collection. V is the set of N normal nodes and n sink nodes, which are deployed randomly in the 2D space. Each normal node collects a specific amount of data $\{\eta_i\}$ in each period, which needs to be transmitted to sinks. The length of a period can range from several seconds to several minutes, depending on the application scenarios. We assume the amount of data collected in each period will not change frequently, therefore we omit the time index of $\{\eta_i\}$. The sink nodes are assumed to be identical. Data is successfully delivered if it reaches any one of the sinks.

Link $\langle i,j \rangle \in E$, if node j is one of the forwarders of i in the next hop. Each node i maintains a set of next-hop forwarders which is denoted as $\mathbb{F}_i$. $\mathbb{F}_i$ contains the neighbors who are within i's communication range and have better routing metrics than node i. In this paper, without loss of generality, the minimum hop count to the sinks is used as the routing metric of a node. Therefore, the DAG represents a minimum hop routing graph in which all sinks are merged into one root node. Each node routes data to sinks via minimum-hop, multi-path routing. We call this scenario as *probabilistic routing under minimum-hop constraint in multi-sink networks*. We focus on the distributed routing probability updating for network lifetime optimization. Note that the presented framework is still valid when other routing metrics are used, only if the acyclic directed graph is provided. We say the nodes with k hops to the sink are *in the kth level*. The DAG is assumed *connected*, i.e., any node in the network can find at least one path to a sink node (Table 1).

2.2 Probabilistic routing model

Each node communicates and learns knowledge only from its one-hop neighbors, which are divided into three groups:

- $\mathbb{S}_i$ are the siblings who has the same hop count with the node i. That is to say, they are in the same level.
- $\mathbb{F}_i$ denotes the forwarders in the *parent level*, which are the neighbors who have better routing process than node i.
- $\mathbb{C}_i$ denotes the children set in the *child level*, which are the neighbors who have i in their forwarder set.

At time t, sensor i collects local data, receives data from children and transmits the aggregated data to forwarders. The routing probability at t is denoted by $\{P_{i,j}(t)\}$, where $j \in \mathbb{F}_i(t)$ and $\sum_{j\in\mathbb{F}_i} P_{i,j}(t) = 1$. Without loss of generality, we assume each sensor initializes its transmission probabilities evenly as $P_{i,j}(0) = 1/|\mathbb{F}_i|$. A sensor can adjust its routing probabilities in each period.

2.3 Energy consumption model

Sensors are assumed have initial energy $\{e_i\}$, which can be either homogeneous or heterogeneous. Previous works [8] also investigated to deploy more powerful nodes close to the sinks to prolong the network lifetime. We consider either the homogenous or heterogeneous sensors obey the following energy consumption model.

Since it is well known that data transmission consumes much more energy than the other modes of a node, we only consider the energy consumed by the radio and assume a sensor consumes one unit of energy for transmitting one unit of message.

In a period t, a sensor receives messages from its children, collects local data and transmits these data to its forwarders. Let $R_i(t)$ be the amount of received data from children; $L_i(t)$ be the amount of data transmitted to its forwarder; Then $L_i(t) = R_i(t) + \eta_i$, where η_i is the locally captured data. Therefore, the energy consumed by sensor i in period t is:

$$L_i(t) = \eta_i + \sum_{j\in\mathbb{C}_i} P_{j,i}(t)L_j(t) \tag{1}$$

where $L_i(t)$ is called the *load* of sensor i in period t. We can see the load depends on the locally captured data, loads of its children and the transmission probabilities from its children to it. The sensors' loads also have an accumulation feature, i.e., the sensors close to the sink generally have higher loads. To prolong the network lifetime, the approaches for load balancing include (i) adding more sinks and placing them evenly in the network; (ii) assigning the nodes close to sinks more power [8]; and (iii) investigating routing algorithms for load balancing. For the fix number of sinks and the fixed initial power of nodes, we investigate the third problem in this paper.

Table 1 List of main notations used in this paper

Notations	Description
$G = (V, E)$	The considered DAG, V is node set; E is edge set
N, n	Number of sensor nodes and number of sinks
η_i	Volume of collected data by i in a round
$P_{i,j}(t)$	Routing probability from i to j at round t
$\mathbb{F}_i$	The forwarder set of node i
$\mathbb{C}_i$	The children set of node i
$L_i(t)$	The load of node i at round t
$C_i(t)$	The local cost of i at round t
$C_{max}(t)$	The maximum local cost of the network at t
e_i	The initial energy of i
$e_i(t)$	The remaining energy of i at t
$\tilde{T}_N^N$	Network lifetime when at least N nodes alive

2.4 Network lifetime model

The load represents the energy consumption rate of a sensor, which has direct impact to the network lifetime. First, at time t, the predicted remaining lifetime of node i can be evaluated by:

$$\tilde{T}_i = t + \frac{e_i(t)}{L_i(t)} \tag{2}$$

where $e_i(t)$ is the remained energy of node i at time t. The network lifetime has various definitions [9], among which the most frequently used one is so called the N-of-N lifetime [6, 10, 11]. In this definition, the network lifetime T_N^N ends as soon as the first node fails, thus

$$\tilde{T}_N^N = \min_{v\in V} \tilde{T}_v \tag{3}$$

where $\tilde{T}_v$ is the predicted lifetime of node v. The advantage of $\tilde{T}_N^N$ is that it is easy to be measured and proper for the worst case analysis. A common variant of the $\tilde{T}_N^N$ metric defines the network lifetime as the time until at least k out of N nodes are alive (k-of-N lifetime $\tilde{T}_N^k$): $\tilde{T}_N^k = \min_{v\in V} \tilde{T}_v$. While this metric is better than $\tilde{T}_N^N$, it lacks the accuracy, since when the fraction of nodes close to sinks are dead, the network is hardly connected, which cannot be revealed in the metric. For quantitative analysis of the network lifetime, we chose network lifetime defined as $\tilde{T}_N^N$.

If the distributed sensors can experience a quick optimization phase such that the transmission probabilities converge to a stable state, a node i can estimate its expected lifetime as its initial energy divided by its expected load, i.e, $\tilde{T}_i(t) = \frac{e_i}{L_i(t)}$, where t is the time index when routing probabilities converge. Thus, the network lifetime optimization problem is to adjust routing probabilities to maximize the minimum expected lifetime of the nodes.

$$\max\{\tilde{T}_N^N\} \Leftrightarrow \max\left\{\min_i\left\{\frac{e_i}{L_i(t)}\right\}\right\}$$
$$\text{s.t.}\, P_{i,j}(t) = P_{i,j}(t-1), \forall\langle i,j\rangle \in E \tag{4}$$

Note $\tilde{T}_N^N$ is actually shorter than the global optimal network lifetime, because the energy consumption is not best balanced before the convergence of the routing probabilities. Therefore, the best network lifetime provided by the distributed probabilistic routing can only be near the optimum. To maximize $\tilde{T}_N^N$, we should not only maximize $\min_i\left\{\frac{e_i}{L_i(t)}\right\}$ at time t, but also minimize t, i.e., making the algorithm converge as fast as possible.

3 Theoretical investigation

We firstly investigate theoretically on the condition for the distributed probabilistic routing to achieve the optimal remaining network lifetime.

3.1 Local cost function

Before presenting the conditions, a distributed cost function should be properly designed, so that each node can adjust their routing probabilities accordingly. Since the problem is essentially to optimize a global objective by sensors' distributed decisions, the local cost should be locally calculated, and be able to imply the load information of the far away ancestors (i.e., forwarders on the paths towards sinks), so that a sensor can make local decision with consideration of global situation.

For the load's accumulation feature, the network lifetime is determined by nodes who have the largest load-energy ratios, which are called *dominating nodes*. The other sensors affect the network lifetime by their routing probabilities to these dominating nodes. Based on this observation, a recursive local cost function is defined as follows:

Definition 1 (*Local Cost*) For a sensor i at round t, its Local Cost (LC) is defined as:

$$C_i(t) = \begin{cases} \dfrac{L_i(t)}{e_i}, & \text{if i is in the first level} \\ \max\left\{\dfrac{L_i(t)}{e_i}, \displaystyle\sum_{j\in\mathbb{F}_i} P_{i,j}(t) * C_j(t)\right\}, & \text{otherwise} \end{cases} \tag{5}$$

Based on the local cost, we give the formal definition of the dominating nodes as:

Definition 2 (*Dominating Node*) Node i is called a dominating node, if it is in the first hop or $\frac{L_i(t)}{e_i} > \sum_{j\in\mathbb{F}_i} P_{i,j}(t) * C_j(t)$; otherwise, it is a non-dominating node.

From (5), we can see that:

1. The LC of each sensor can be computed locally by only the knowledge of the LCs of its forwarders, the routing probabilities to the forwarders, and the load-energy ratio of itself;
2. The LC is a recursive function of the dominating nodes' load-energy ratios and the transmission probabilities along the paths to the dominating nodes;
3. If a node is a non-dominating node and all its forwarders have the same LC, it will have the same LC as its forwarders, because $\sum_{j\in\mathbb{F}_i} P_{i,j}(t) = 1$.

3.2 Sufficient condition for optimizing remaining lifetime

Based on the local cost function, we have:

Theorem 1 (Sufficient condition) *For a data collection sensor network running distributed probabilistic routing, the remained network lifetime is optimal if all the sensors in the network have the same local costs.*

Proof When all the sensors have the same LC, the sensors in the first level must have the same energy-load ratio. Supposing there are K nodes in the first level, we have $\frac{e_1}{L_1(t)} = \frac{e_2}{L_2(t)} = \ldots = \frac{e_K}{L_K(t)} = \mathscr{T}$. Since all data need to be forwarded to sinks by the first level nodes, by rearrangement inequality [12], any modification of routing probability can only reduce the network lifetime. This is because the modification will cause unbalance of the load distribution of the first level, which can only reduce the network's lifetime. □

However, in practice, sensors have limited transmission range and the network is generally randomly deployed. The network topology constraint may prevent the satisfactory of above sufficient condition. An extreme case is that if a sensor has only one forwarder, it can not adjust routing probability. An example is shown in Fig. 1. The local costs of the sensors can not reach equality because of the topology constraint.

3.3 Local condition for optimizing remaining lifetime

We need to find a condition which can be evaluated by sensor distributively to verify whether it has reached the optimal condition for network lifetime optimization. Taking the topology constraint into account, a distributed condition is stated as following:

Theorem 2 (Sufficient distributed condition) *The remaining network lifetime is optimal at time t, if for any sensor i, it satisfies the following local routing probability assignment conditions*:

1) If i is a non-dominating node,

- $P_{i,j}(t) > 0$, *for* $j \in \mathbb{F}_i$ *and* $C_j(t) = C_i(t)$;
- $P_{i,j}(t) = 0$, *for* $j \in \mathbb{F}_i$ *and* $C_j(t) > C_i(t)$;
- *there is no topology link from i to nodes in the forwarder set* $\mathbb{F}_i$ *with LC smaller than i.*

2) *If i is a dominating node,*

- $P_{i,j}(t) > 0$, *if for* $j \in \mathbb{F}_i$, *j is a sink or* $C_j(t) \leq C_i(t)$;
- $P_{i,j}(t) = 0$, *for* $j \in \mathbb{F}_i$ *and* $C_j(t) > C_i(t)$.

Proof We prove this distributed condition by two steps: the first step proves the network will form into some cost equivalent sets and the second proves the remaining network lifetime is optimal.

STEP1: From Condition (1), for any non-dominating node i, if $P_{i,j}(t) > 0$, there must be $C_j(t) = C_i(t)$. The fact holds for its forwarders and children. Therefore, these non-dominating nodes with the same LC form one cost-equivalent set. Among different cost equivalent sets, the non-dominating nodes with the larger LC have no topology links to the node with the smaller.

From Condition (2), only the dominating node in each set can have positive links to the sink or to the other nodes with smaller LCs. So the dominating nodes connect different cost equivalent sets.

STEP2: *We will prove the remaining network lifetime is optimal in the network structure stated in STEP1.* Note that the nodes with the overall maximum LC must be in the same cost-equivalent set. We denote this set as $\mathbf{M_{max}}$. From condition (1), all the non-dominating nodes in set $\mathbf{M_{max}}$ have no topology links to nodes in other sets with smaller LC. Only the dominating nodes in this set have positive links to the other sets. From (5), the LC of nodes in this set is determined by the dominating nodes. Since all non-dominating nodes in this sets have no links to the other sets, there is no way for this set to reduce its total load. Since all the loads from this set must be forwarded by the dominating nodes to sinks or other sets, from Rearrangement Inequality [12], any rearrangement of the routing probabilities in the set can only increase the load-energy ratios of some dominating nodes. Therefore, the network lifetime has reached the optimum. □

From the above theorem and its proof, we can have the following corollaries.

Corollary 1 (Detectability of distributed condition) *The condition stated in Theorem 2 is detectable and can be verified by each sensor distributively.*

Proof Each node only need to determine whether it is dominating or non-dominating by (5), and checks the conditions in Theorem 2. □

Corollary 2 (Cost-equivalent sets) *When nodes have optimally assigned their routing probabilities for remaining network lifetime optimization, the network has self-organized into some cost-equivalent sets. In each cost-equivalent set, only the dominating nodes have positive links to the nodes in other sets.*

Corollary 3 (Non-uniqueness) *The optimal routing probability assignments are not unique.*

Proof An example is constructed to prove the non-uniqueness. As shown in Fig. 1(b), all sensors have best assigned their routing probabilities for network lifetime optimization. In the set with $LC = 5$, the two leaf nodes can select numerous combinations of transmission probabilities, such as [0.9,0.1], [0.1,0.9]. These different probability assignments all lead to the same LCs of forwarders. □

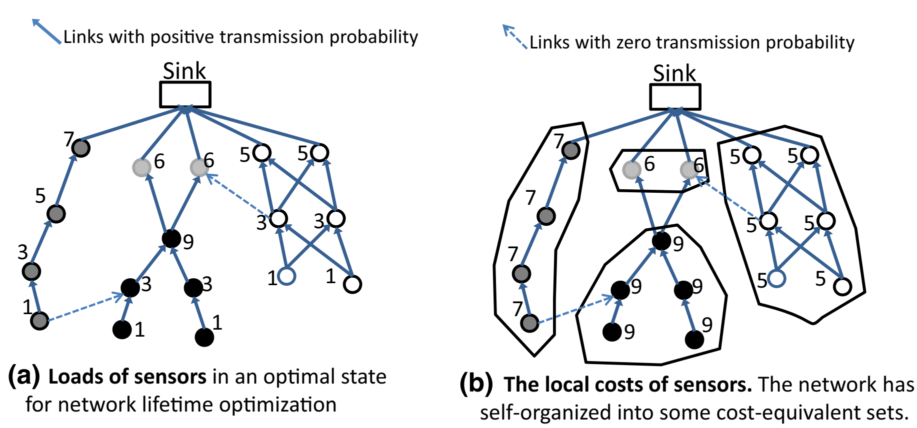

Fig. 1 The loads and local costs of sensors in an optimal state for the network lifetime optimization. The nodes have self-organized into some cost-equivalent sets

In Fig. 1, we can see the nodes form four cost-equivalent sets. The dashed links have zero transmission probabilities, so that the non-dominating nodes in different cost-equivalent sets are isolated. Figure 1(a) shows the loads of the sensors in this scenario. The network lifetime is determined by the node with LC equal to 9, which is the dominating node in $\mathbf{M}_{\mathbf{max}}$ in Fig. 1(b).

4 Distributed adaptive probabilistic routing

The conditions in Theorem 2 provide directions to design the distributed probabilistic routing algorithm for network lifetime optimization. Any sensor should self-optimize its transmission probabilities locally to balance the local cost of its forwarders. However, the distributed cost balancing problem is still challenging, because of the *load oscillation (instability)* and the problem of falling into *local optimum*.

4.1 Challenges in local cost balancing

4.1.1 Load oscillation

Load oscillation problem happens when each node tries its best to balance its forwarders' loads. Because a sensor is unaware of the other siblings' concurrent decisions, so the close-by siblings may react similarly and concurrently due to their observations to the loads of their common forwarders. This may lead to overreaction, i.e., the loads of their common forwarders are changed from over-high to over-low and vice versa. An example of the load oscillation problem is shown in Fig. 2(a) when running the greedy algorithm in [13] for load balancing. The loads of sensors in the network oscillate between state 'A' and 'B' but can not converge.

4.1.2 In-level local optimum problem

In-level local optimum problem means the sensors of a child level may fail to balance the loads of their parent level because of falling into local optima. For example, using the distributed load-balanced tree strategy (DLBT) in [14], we can verify that the routing probabilities of the sensors in the second level in Fig. 2(b) has converged. However, the loads in the first level are not balanced.

4.1.3 Over-level local optimum problem

Besides the in-level local optimum problem, there is the *over-level local optimum problem*, which means even if the sensors in each level have optimally balanced loads of their parent level, the network lifetime is still not optimized. The reason is that in order to achieve network lifetime optimization, some levels have to sacrifice their local balance to minimize the load of the dominating node. An example is illustrated in Fig. 2(c). The network on the left is in a cross-level local optimum state. Because children make routing decisions earlier than their forwarders (forwarders can adjust routing decision only after receiving data from children), in the figure all sensors have tried their best to balance the loads of their parent level. But we can see the load of the first level is not balanced. The right figure gives the globally optimal state of the network. To maximize the lifetime, it requires the load distribution in the second level to be unbalanced. We resolve such problems by the balancing of local costs.

4.2 Guiding policy for distributed probabilistic routing

In order to tackle above oscillation and local optimum challenges, we present a virtual guiding policy to guide sensors to update their local transmission probabilities towards the optimal state in Theorem 2. Let's define $E(C)$ as the *virtual guidance*, which indicates an ideal balanced LC of sensors by assuming a virtual state in which all sensors have already balanced their LCs.

In probabilistic routing, if a sensor can know the value of $E(C)$, it would have a guidance for updating its local transmission probabilities. Each node would try to balance its LC to approach $E(C)$, which will avoid the instability and local optimum problems.

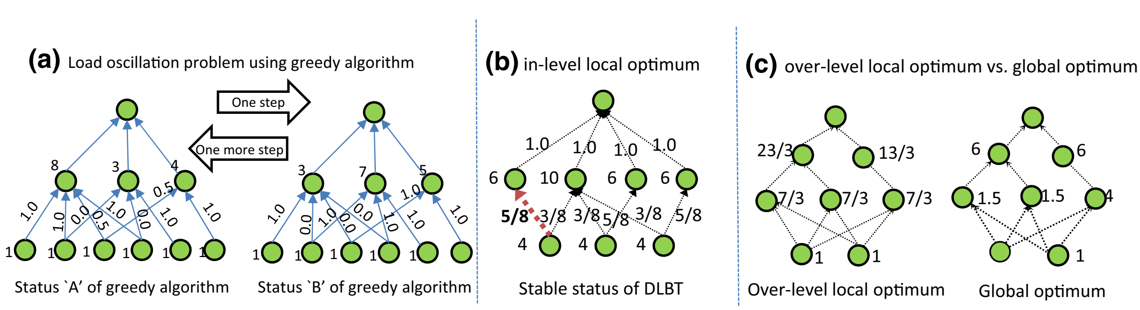

Fig. 2 Example of over-level local optimal versus the global optimal scenarios

For a child node i, suppose $j \in \mathbb{F}_i$ is one of its forwarders. $C_j(t-1)$ is the LC of j in round $t-1$. In round t, when node i updates the transmission probability $P_{i,j}(t)$, it is virtually guided to balance j's LC towards $E(C)$:

$$P_{i,j}(t) = P_{i,j}(t-1)\frac{E(C)}{C_j(t-1)M_i(t-1)}, \tag{6}$$

where $M_i(t-1) = \sum_{j \in \mathbb{F}_i} P_{i,j}(t-1)\frac{E(C)}{C_j(t-1)}$ is a normalizer that keeps the sum of the transmission probabilities of node i equal to 1. By substituting $M_i(t-1)$ into Eq. (6), we have

$$P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j \in \mathbb{F}_i} P_{i,j}(t-1)/C_j(t-1)}, \tag{7}$$

Note that $E(C)$ is counteracted in Eq. (7). $\{C_j(t-1)\}$ and $\{P_{i,j}(t-1)\}$ can be learned from forwarders. This enables a distributed routing probability updating scheme at each distributed sensor.

4.3 Distributed adaptive probabilistic routing (DAPR)

Based on the virtually guided routing probability updating in Eq. (7), we propose a DAPR algorithm to optimize the remaining network lifetime, which handles: (1) transmission scheduling; (2) neighborhood information exchanges; and (3) routing probability updating.

1. Transmission and Updating Scheduling Data is collected round by round. In each round, each sensor adjusts its routing probabilities before transmitting data. More specifically, in each round sensors in the outmost level collect and transmit data firstly. Sensors in level k transmit data after receiving data from children in level $k+1$. Therefore, in each round sensors in level k adjust routing probabilities after the sensors in level $k+1$.

2. Information Exchange In any round, each sensor adjusts routing probability from $P_{i,j}(t-1)$ to $P_{i,j}(t)$ $j \in \mathbb{F}_i$ based on the neighborhood information given in Table 2. Each sensor needs to update its local information in each round. Therefore, when a sensor i transmits data packet to a forwarder j, its state information $\{L_i(t), C_i(t), P_{i,j}(t)\}$ is embedded into the preamble of its packet. When a sensor receives a packet from a child, it obtain the child's state information at t. When a sensor overhears a packet from its forwarder, it updates the state information of the forwarder.

3. DAPR Algorithm Based on above scheduling and information exchanging scheme, the DAPR algorithm for sensor i in round t is listed in Algorithm1. In the algorithm, from line 2 to line 4, sensor i updates state information of its children. Then it updates its load by line 5 and adjusts routing probabilities to its forwarders by the virtually guiding method in lines 6–10. Then it transmits data to its forwarders. At line 12, when its forwarders transmit data, it overhears the packet to update the local cost of the forwarders at time t.

Algorithm 1 DAPR Algorithm for a sensor i, in round t

1: Node i captures its local data η_i.
2: Node i receives data from children.
3: Learn the current loads of children : $\{L_m(t) | m \in \mathbb{C}_i\}$
4: Learn the children's transmission probabilities: $\{P_{m,i}(t) | m \in \mathbb{C}_i\}$
5: Estimate its own load: $L_i(t) = \eta_i + \sum_{m \in \mathbb{C}_i} P_{m,i}(t)L_m(t)$
6: Denote $j = \mathbb{F}_i[j]$ as the jth forwarder of node i.
7: **for** $(j = 1; j \leq |\mathbb{F}_i|; j++)$ **do**
8: $\quad P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j \in \mathbb{F}_i} P_{i,j}(t-1)C_j(t-1)}$
9: $\quad C_i(t) = \begin{cases} L_i(t)/e_i, & \text{if } i \text{ is in the first level} \\ \max\left\{\frac{L_i(t)}{e_i}, \sum_{j \in \mathbb{F}_i} P_{i,j}(t) * C_j(t-1)\right\}, & \text{otherwise} \end{cases}$
10: **end for**
11: Transmit data to forwarders by $\{P_{i,j}(t) | j \in \mathbb{F}_i\}$.
12: Overhear data of forwarders to update the local cost of forwarders: $\{C_j(t), j \in \mathbb{F}_i\}$.

Table 2 Local information of i used for routing probability updating

$L_m(t), m \in \mathbb{C}_i$:	The loads of children at t
$P_{m,i}(t), m \in \mathbb{C}_i$:	Routing probabilities from children to it at t
$P_{i,j}(t-1), j \in \mathbb{F}_i$:	Routing probabilities to forwarders at $t-1$
$C_j(t-1), j \in \mathbb{F}_i$:	Local cost of forwarders at $t-1$
η_i:	Amount of locally collected data

5 Performance evaluation

In this section, we will analyze the performance of our DAPR algorithm, such as its convergence, providing the optimal remaining network lifetime, and its convergence speed.

5.1 Convergence property of DAPR

We firstly prove the convergence of DAPR by three steps.

1. The convergence problem is reduced to a convergence problem in two-level networks.
2. In the two-level network, we prove the maximum LC of all the nodes in the parent level (denoted as $C_{max}(t)$) will converge.
3. When $C_{max}(t)$ converges, we prove the convergence property of DAPR by subtraction and reconstruction.

Lemma 1 **(Reduce to two-level network problem)** *The convergence of DAPR in a multi-level DAG network is identical to the convergence problem of DAPR in the outmost two-levels.*

Proof In DAPR, sensors in level $k+1$ always adjust routing probabilities earlier than sensors in level k. From Eqn. (1), when the loads and routing probabilities in level $k+1$ converge, the loads of the nodes in level k will converge automatically. Because the loads of the outmost level are initially stable, if the routing probabilities from the outmost level nodes to their forwarders converge, their forwarders' loads will become stable. Therefore, if the routing probabilities between the outmost two levels can converge to be stable, the stability will eventually propagate to the whole network. □

Thus, we can consider stability problem in a two level network, which consists of a *child level* and a *parent level*. The loads of the *child level* are initially stable.

Lemma 2 (Convergence of $C_{max}(t)$) *Let $C_{max}(t)$ be the maximum local cost of the nodes in the parent level. For a two level network running DAPR, $C_{max}(t)$must be decreased continually until $C_{max}(t) = C_{max}(t-1)$.*

Proof Consider the two scenarios in the definition of the LC in Eq. (5):

Case 1: if i is a nondominating node, $C_i(t) = \sum_{j\in\mathbb{F}_i} P_{i,j}(t)\cdot C_j(t-1)$, **then:**

$$\begin{aligned}
C_i(t) &= \sum_{j\in\mathbb{F}_i} P_{i,j}(t)\cdot C_j(t-1),\\
&= \sum_{j\in\mathbb{F}_i}\left(\frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j\in\mathbb{F}_i} P_{i,j}(t-1)/C_j(t-1)}\cdot C_j(t-1)\right)\\
&\le \sum_{j\in\mathbb{F}_i}\left(\frac{P_{i,j}(t-1)}{\sum_{j\in\mathbb{F}_i} P_{i,j}(t-1)/C_{max}(t-1)}\right)\le C_{max}(t-1)
\end{aligned} \tag{8}$$

Case 2: if i is a dominating node, $C_i(t) = L_i(t)/e_i$, **then:**

$$\begin{aligned}
C_i(t) &= \frac{u_i}{e_i} + \sum_{m\in\mathbb{C}_i}\frac{P_{m,i}(t)L_m(t)}{e_i},\\
&\le \frac{u_i}{e_i} + \sum_{m\in\mathbb{C}_i}\frac{\frac{P_{m,i}(t-1)}{C_i(t-1)}}{\sum_{k\in\mathbb{F}_m}\frac{P_{m,k}(t-1)}{C_{max}(t-1)}}\cdot\frac{L_m(t-1)}{e_i}\\
&= \frac{u_i}{e_i} + \frac{C_{max}(t-1)}{C_i(t-1)}\sum_{m\in\mathbb{C}_i}\frac{P_{m,i}(t-1)L_m(t-1)}{e_i}\\
&\text{because}\sum_{m\in\mathbb{C}_i}\mathrm{P_{m,i}(t-1)L_m(t-1) = L_i(t-1) - u_i}\\
C_i(t) &\le \frac{u_i}{e_i} + \frac{C_{max}(t-1)}{C_i(t-1)}\frac{(L_i(t-1)-u_i)}{e_i}\\
&= C_{max}(t-1) - \left(\frac{C_{max}(t-1)}{C_i(t-1)} - 1\right)\\
&\frac{u_i}{e_i} \le C_{max}(t-1)
\end{aligned} \tag{9}$$

It should be noted that in Eq. (9), $L_m(t) = L_m(t-1)$ because the child level sensors have stable loads. Therefore, since $C_{max}(t) = \max\{C_i(t)\}$, we get

$$C_{\max}(t) \le C_{\max}(t-1). \tag{10}$$

From Eqs. (8) and (9), $C_{max}(t)$ is equal to $C_{max}(t-1)$ only if: (i) all the parent level sensors have local cost equal to $C_{max}(t-1)$, or (ii) the parent nodes have no children with local cost less than $C_{max}(t-1)$. In both cases, $C_{max}(t) = C_{max}(t-1)$, which has reached a stable value. Otherwise, $C_{max}(t)$ will be decreased continuously until $C_{max}(t) = C_{max}(t-1)$. Since $C_{max}(t)$ is lower bounded which cannot be smaller than the average local cost of the parent level, this implies that $C_{max}(t)$ will be reduced continuously until convergence. Therefore, Lemma 1 is proved. □

When $C_{max}(t)$ converges, there is at least one node in the parent level has local cost equal to $C_{max}(t)$. We call such nodes *the maximum cost nodes* (*MCNs*). Analyzing the children of MCNs, we can find their transmission probabilities to the MCNs have following properties.

Lemma 3 **(Local structure when $C_{max}(t)$ converges)** *When $C_{max}(t)$ is stable,*

- *if a node in the child level has LC equal to $C_{max}(t)$, all its forwarders must be MCNs; and*
- *if the node has LC less than $C_{max}(t)$, but has some MCNs in its forwarder set, its transmission probabilities to these MCNs must be* 0.

Proof The two parts of Lemma 3 are proved separately. Note that the equality condition in Eq. (8) holds when

$$\sum_{j \in \mathbb{F}_i} P_{i,j}(t-1)/C_j(t-1) = 1/C_{max}(t-1). \tag{11}$$

It means $C_i(t) = C_{max}(t-1) = C_{max}(t)$. So, only when the local costs of all the forwarders of i are equal to $C_{max}(t-1)$, can Eq. (11) hold. Therefore, if a child node has local cost equal to $C_{max}(t)$, all its forwarders are MCNs.

To prove the second part of Lemma 3, suppose a child i has local cost $C_i(t) < C_{max}(t)$, and it has a parent j which is a MCN. By Eq. (7), we have

$$\frac{P_{i,j}(t)}{P_{i,j}(t-1)} = \frac{1/C_{max}(t-1)}{\sum_{m \in \mathbb{F}_i} P_{i,m}(t-1)/C_m(t-1)} < 1. \tag{12}$$

As time passes, the transmission probability from i to j will decreases until $P_{i,j}(t) = P_{i,j}(t-1)$, i.e.,

$$P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_{max}(t-1)}{\sum_{m \in \mathbb{F}_i} P_{i,m}(t-1)/C_m(t-1)} = P_{i,j}(t-1). \tag{13}$$

The only condition Eq. (13) holds is that $P_{i,j}(t) = P_{i,j}(t-1) = 0$. Therefore, the transmission probability from i to j will decrease monotonously to 0. □

Figure 3 illustrates the scenario for better understanding of Lemma 3. After the convergence of $C_{max}(t)$, for a child j which has MCNs in its forwarder set, either its LC is equal to $C_{max}(t)$, or its LC is less than $C_{max}(t)$ and it assigns zero transmission probabilities to the MCNs .

Based on the above lemmas, we can arrive at the convergence property of DAPR.

Theorem 3 (Convergence property) *For a data collection network running DARP algorithm, as time passes, the routing probabilities and the local costs of all sensors must converge to stable values.*

Proof Theorem 3 is proved by an isolation and reconstruction method. Lemma 2 have proved that after $C_{\max}(t)$ converges, if a child node has local cost equal to $C_{\max}(t)$, all its forwarders must be MCNs; if a node i has local cost $C_i(t) < C_{\max}(t)$, its transmission probabilities to these MCN forwarders must converge to zero (Fig. 3).

Therefore, the MCNs and their children who have positive links to them form an isolated set in the network, i.e., *the network can be divided into a MCN set and a non-MCN set. The transmission probabilities between the MCN set and the non-MCN set are zero*. Therefore the nodes in the MCN set have no effects to the non-MCN set. If we subtract the nodes and links in the MCN set, the local costs and the transmission probabilities among the nodes in the remained non-MCN set will not be affected.

After the subtraction, Lemmas 2 and 3 will still hold for the remaining network, and a new set of MCNs can be determined in the remaining network. We repeat the subtraction process until all the remaining nodes form one MCN set, in which all the nodes left have the equal local costs. In such a case, it is easily verified by (7) that $\forall i, j$ $P_{i,j}(t) = P_{i,j}(t-1)$. So, it is a stable state. Then by adding the removed MCN sets and links back into the network in the reversed order, the stable scenario of the network is obtained. □

Using the convergence property of the two-level networks repeatedly, we can prove that DAPR can guarantee the convergence of the transmission probabilities in multi-hop data collection networks.

5.2 Optimality property

Theorem 4 (Optimality property) *In the convergence state of DAPR, each sensor's routing probability assignment satisfies the local condition specified in Theorem 2, and therefore is optimal for maximizing the remaining network lifetime.*

Proof

1. The subtraction and reconstruction process stated in the proof of Theorem 3 will generate a number of cost-equivalent sets. Nodes in each cost-equivalent set have the same local cost.
2. Lemma 2 guarantees that a non-dominating node has no link to nodes with smaller LC than itself and its

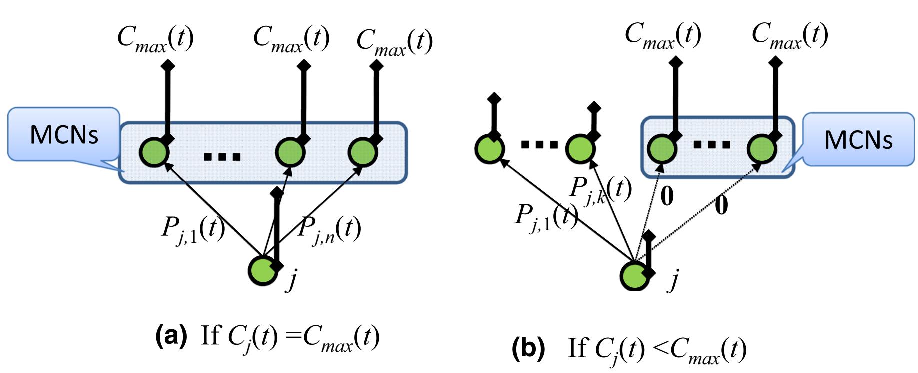

Fig. 3 An example to explain the state when $C_{max}(t)$ is converged

transmission probabilities to the forwarders with higher LCs have converged to zero.

Therefore the stable state of DAPR has met the conditions stated in Theorem 2 for optimizing the remaining network lifetime. □

5.3 Convergence speed

Fast convergence is important for optimizing the network lifetime, which can avoid the unbalanced energy consumption during the route optimization process. To evaluate the convergence speed of DAPR, let's consider an arbitrary node i and suppose it has K forwarders, i.e., $|\mathbb{F}_i| = K$. When $C_1(t-1) = C_2(t-1) \cdots = C_K(t-1)$, $\forall j \in \mathbb{F}_i, \frac{P_{i,j}(t)}{P_{i,j}(t-1)} = 1$. So, we only need to consider the convergence speed when the forwarders' costs are not balanced.

Assume the K forwarders are divided into k sets $\mathbf{X_1}$, $\mathbf{X_2}$,..., $\mathbf{X_k}$ following a descending order of the local costs at time $t-1$. Let's set $\frac{\max_{j \in \mathbb{F}_i}\{C_j(t-1)\}}{\max_{j \in \mathbb{F}_i \setminus \mathbf{X_1}}\{C_j(t-1)\}} = 1 + \Delta_1$. Here, we define $\Delta_a > 0 (1 \le a < k)$ to indicate the local cost difference between the set $\mathbf{X}_a$ and $\mathbf{X}_{a+1}$. Δ_1 indicates the local cost difference between the largest and the second largest local cost set of the forwarders.

Let m be one of the forwarders that has the largest local cost, the convergence speed of $P_{i,m}(t)$ can be derived to have the following pessimistic bound.

$$\frac{P_{i,m}(t)}{P_{i,m}(t-1)} \le \frac{1}{1 + \left(1 - \sum_{j \in \mathbf{X_1}} P_{i,j}(t-1)\right)\Delta_1} \tag{14}$$

Set $1 - \sum_{j \in \mathbf{X_1}} P_{i,j}(t-1) = \beta_1 > 0$. $P_{i,m}(t)$ decreases towards zero in the speed of $\frac{1}{1+\beta_1\Delta_1}$ in one step, which is quick when Δ_1 is large. When $P_{i,m}(t) * *(\mu \in \mathbf{X_1})$ is close to zero, we can assume $\mathbf{X_1}$ is removed from the forwarder set. The transmission probabilities from i to the remained largest local cost forwarder set $\mathbf{X_2}$ will converge following the same way. Note that if Δ_a is very small, $\mathbf{X}_a$ and $\mathbf{X}_{a+1}$ can be thought as in the same set. The convergence speed will be quick until the remained forwarders all have the similar local costs, which is close to the optimal state. At such a near optimal state, since $\frac{1}{1+\beta_a\Delta_a}$ is close to 1, the convergence becomes slow.

Therefore, the convergence characteristics of DAPR are:

- The routing probabilities converge to a near optimal state quickly; and
- evolve towards the optimum state slowly after the local costs of the forwarders are almost balanced.

5.4 Enhance the convergence speed of DAPR

Based on the convergence speed characteristics, we further propose a Fast-DAPR to (1) speed up the convergence speed when routing probabilities are near optimal; and (2) to help DAPR adapt quickly to online network dynamics.

The reason of slow convergence when the network is close to the optimal state is due to Eq. (7). Since $P_{i,j}(t) = \frac{P_{i,j}(t-1)/C_j(t-1)}{\sum_{j \in \mathbb{F}_i} P_{i,j}(t-1)/C_j(t-1)}$, when $P_{i,j}(t-1)$ is almost zero, $P_{i,j}(t)$ obtains very small gain in every period. To speed up the convergence speed, two strategies are designed for DAPR, called Fast-DAPR.

- Rule 1: Let $\varepsilon_1 > 0, \varepsilon_2 > 0$ be two positive small thresholds. At time t, if sensor i discovers a forwarder j has local cost larger than itself, i.e., $C_j(t) > C_i(t)$, $P_{i,j}(t) < \varepsilon_1$, and $\frac{P_{i,j}(t)}{P_{i,j}(t-1)} > 1 - \varepsilon_2$, then node i assigns $P_{i,j}(t)$ directly to zero.
- Rule 2: From the stable state, if a sensor i detects a forwarders j suddenly has smaller local cost than itself, i.e., $C_j(t) < C_i(t)$ but $P_{i,j}(t) \approx 0$, then it resets the transmission probabilities to all forwarders evenly and run the Fast-DAPR algorithm to readjust the transmission probabilities.

We will show in the next section, by running Fast-DAPR, sensors can quickly self-optimize the routing probabilities to lead the network to a stable and lifetime-optimal state. Moreover, from the stable state, the sensors adapt to the network topology and the data flow dynamics quickly.

6 Simulation results

We conduct simulations and develop a demo to evaluate the DAPR algorithm [15], including its convergence, optimality, self-adaptivity and the convergence speed.

6.1 Setups

Simulations were conducted by Matlab2011 using the Directed Adjacent Graph (DAG) library [16] to model a multi-hop network. We modified the weights of the links in the DAG to model the transmission probabilities. The communication radius r of the sensors are identical and normalized as $r = 1$ unit, where a unit is generally 50 m in practice. All the sensors' initial energy is set to 1 unit, where a unit is generally one microwatt. For dynamic networks, we consider the changes of the network topology and the data collection rates.

Four other algorithms are evaluated and compared:

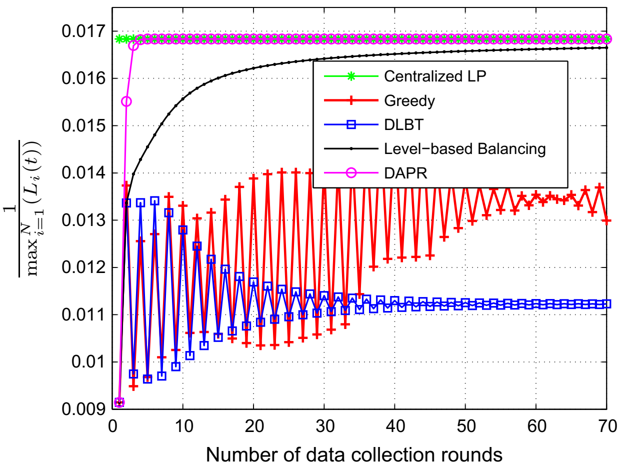

Fig. 4 Performances in a high-density static network

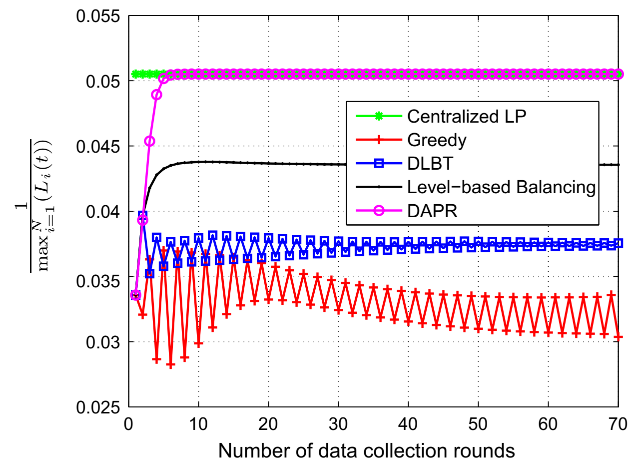

Fig. 5 Performances in a low-density static network

- Centralized Linear Programming (LP) using global information [6], which provides the global optimal lifetime;
- Greedy algorithm for local load-balancing (Greedy), i.e., each sensor tries on its own to best balance the loads of its forwarders;
- Dynamic Load-Balanced routing Tree (DLBT) [14];
- Level-based load balancing algorithm (Level-based Balancing [13]).

6.2 Performance in static networks

We first evaluate the convergence and optimality of DAPR in static networks. We evaluate DAPR in both high and low density networks. In a high-density network, 300 nodes are randomly deployed in a $7r \cdot 7r$ area. In a low-density network, 100 nodes are randomly deployed in a $10r \cdot 10r$ area. The maximum real-time energy-load ratio of all sensors, i.e., $\frac{1}{\max_{i=1}^{N}(L_i(t))}$ is plotted as the evaluation metric of the network lifetime for the different algorithms. Figure 4 shows the performance of the different algorithms in the high density network, and Fig. 5 shows their performance in the low density network.

From the results we can see DAPR leads the network to a stable and optimal state for network lifetime optimization. In the figures, the green line is the optimal network lifetime given by centralized linear programming. For the networks of different densities, DAPR can always converge to a stable state very close to the global optimal performance. The greedy algorithm shows instability because of its zigzaging performance curves. The DLBT algorithm and the Level-based Balancing converge to local optimal results.[1]

6.3 Convergence speed in different networks

The evaluation results showed that the DAPR algorithm generally converges quickly (within several rounds) from the initial state to the global optimum state in different networks. It should be noted that in the initial state all the nodes assign even transmission probabilities to their forwarders. Later, after introducing the adaptivity in the dynamic networks, we will show that the initial state is an important reason for the quick convergence.

6.4 Performance in dynamic networks

We evaluate the self-adapting and convergence performance of DAPR under two kinds of dynamic events:

1. Topology Dynamics: the network's topology changes;
2. Flow Dynamics: the amount of data collected by sensors changes over time.

If the dynamic events are highly frequent, it will be hard for any algorithm to achieve online adaptation, especially when the interval between the dynamic events is shorter than the convergence time. Therefore, we focus on the evaluations of DAPR in networks with not highly frequent dynamical events.

6.4.1 Self-adaptation to topology dynamics

Topology dynamics include joining, leaving and moving of nodes. When a new node joins the network, it will discover its forwarders and child candidates by local communications. Then, it assigns transmission probabilities evenly to

[1] In our online demo [15], users can generate random network topologies or manually control the network topologies to evaluate and

Footnote 1 continued
compare the performance of DAPR, Greedy, DLBT and Level-based balancing algorithms.

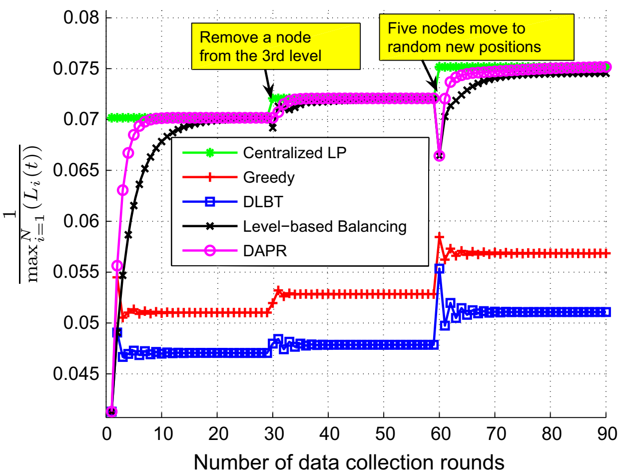

Fig. 6 Self-adapting to topology dynamics

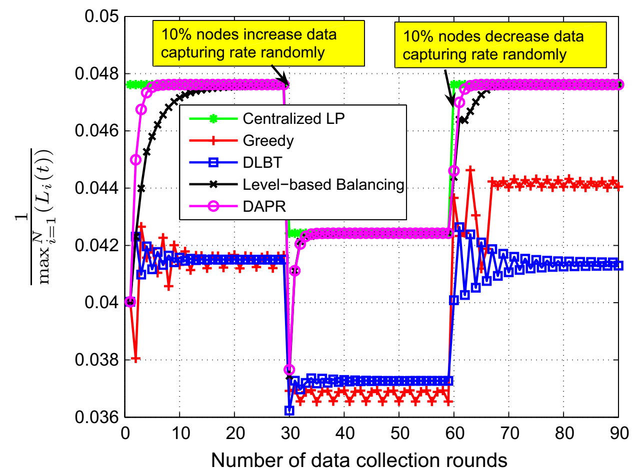

Fig. 7 Self-adapting to flow dynamics

the forwarders and notifies the children to reset their transmission probabilities. Node leaving is handled similarly. The children of the leaving node will reset its transmission probabilities to exclude the leaving node. The forwarders of the leaving node will also notify their children to adjust their transmission probabilities. Node movements are treated as a merge of leaving and joining.

Figure 6 shows how DAPR quickly adapts to topology dynamics in a network with 100 nodes and eight levels. In the 30th round when five randomly chosen nodes are removed, DAPR leads all the sensors to converge to a new stable optimal state quickly. In the 60th round, when another five nodes are moved to some random new positions, the network self-adapts and converges to a new stable state after only several rounds.

6.4.2 *Self-adaptation for flow dynamics*

DAPR can also self-adapt to flow dynamics as the example shown in Fig. 7. The network has 100 nodes, which converges to a stable state after about 5 rounds. After that, in the 30th round, 10 % nodes increase data capturing rates randomly. DAPR uses about 5 rounds to converge to a new stable optimal state. In the 60th round, when 10 % nodes decrease data capturing rates randomly, DAPR uses about 5 rounds to converge again.

Extensive simulations show that DAPR generally self-adapt to the topology and network flow dynamics quickly, but there are some special cases. We show these special cases can be well resolved by using Fast-DAPR algorithm.

6.5 Slow adaptation cases and fast-DAPR algorithm

Slow adaptation happens when a node need to update its routing probability from near zero to a bigger value. The convergence process of $\frac{1}{\max_{i=1}^{N}(L_i(t))}$ in an evaluated network is plotted in Fig. 8. It can be seen that it converges to the first stable state quickly. But when a node whose children has routing probability nearly 1 to it is deleted, the convergence is slow. This is because the children need to adjust the routing probability to other forwarders from near zero to some positive values.

For the same event, when we run the Fast-DAPR algorithm, the variation of $\frac{1}{\max_{i=1}^{N}(L_i(t))}$ is plotted in Fig. 9. Comparing with Fig. 8, we can see the adaptation speed has been remarkably improved. This is because the resetting operation in Fast-DAPR avoids the routing probability to be adjusted from near zero, which speeds up the convergence. The observation also explains the fast convergence of DAPR in the static networks, because in the initial state all nodes have even transmission probabilities, which is a good starting state for fast convergence of DAPR.

7 Related work

We review related works from five aspects: (1) centralized approaches, (2) distributed approaches, (3) probabilistic routing algorithms, and (4) others.

1. Centralized approaches: The lifetime optimal routing problem was studied using linear optimization. Buragohain, et al. [6] studied the optimal routing tree, which proved it is NP-Complete to find the energy-optimal routing tree for *partially aggregated* and *unaggregated* queries in random networks. They claimed probabilistic routing can improve the network lifetime as opposed to the tree-based approaches. They proposed a centralized linear programming model to solve the network lifetime maximization problem by probabilistic routing. In [17], Chang

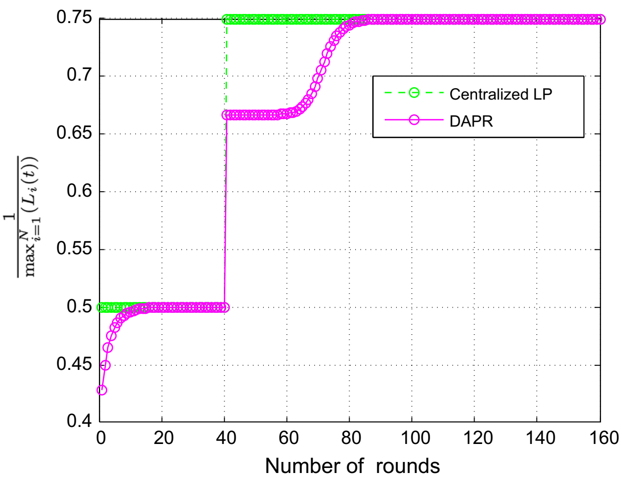

Fig. 8 Self-adapting to the topology changing event using DAPR

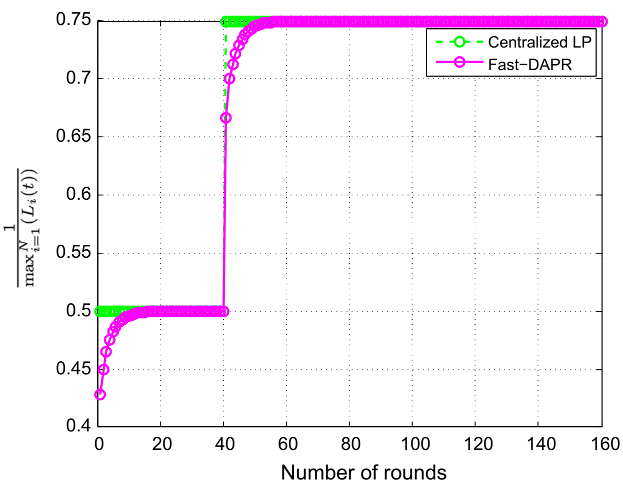

Fig. 9 Self-adapting to the topology changing event using Fast-DAPR

et al.. proposed the Garg-Koenemann algorithm and provide a centralized combinatorial approximation solution. In [18], Sankar et al. proposed linear optimization approach based on the formulation of multicommodity flow. They also proposed a distributed routing algorithm that has asymptotically small relative error with the centralized algorithm. Liang and Liu [10, 11] studied network lifetime maximization through constructing an energy optimized spanning tree. They proved the NP-completeness of the problem, and devised several heuristics to prolong the network lifetime. Dai, et al. [19] studied the load balancing problem in a grid topology. They made use of the Chebyshev sum inequality as a criterion for load balancing.

2. Distributed approaches: Distributed algorithms were proposed to solve the maximum lifetime routing problem. In [20], Madan et al. formulated the maximum lifetime routing as a linear programming problem and proposed distributed subgradient algorithms. In [21] authors regarded the lifetime as resources to propose a utility-based nonlinear optimization formulation.

In another category of distributed probabilistic routing, Powell, Efthymiou and Jarry et al. [3–5] studied probabilistic algorithms to decide in each step whether to propagate the data to the one-hop forwarders, or send data directly to the sink. This randomized choice balances the (cheap) one-hop transmissions with the (expensive) direct transmissions to the sink, which was formally shown to best balance the energy consumption rates of the sensors. Jarry et al. [22] further proved that the network lifetime maximization is equivalent to the flow maximization problem in data collection networks and presented distributed algorithms to solve it. A more recent work by Boukerche et al. [8] studied distributed probabilistic routing based on limited local network density information under both uniform and heterogeneous sensor deployments. Their algorithm can achieve energy balance while minimizing energy consumption. However, a general assumption in these studies is that the sensors can transmit data to the nodes beyond the parent level or directly to the sink, which requires sensors have long and adjustable communication ranges.

Many other distributed routing algorithms are based on heuristic approaches. Parametric Probabilistic Routing (PPR) [23] addressed the energy efficient routing by assigning higher transmission probabilities to the nodes closer to the destination. The Dynamic Load Balance Tree (DLBT) algorithm [14] calculated routing probabilities using a heuristic in which the transmission probabilities should be inverse-proportional to the corresponding forwarders' load. However, the stable state of DLBT may fall into level-type local optimum (see the explanation in Section II). Wang et al. [13] presented a level-based load balancing algorithm to achieve level-by-level load balancing in data collection networks, which may fall into over-level local optimum. In [24], Chen et al. presented a distributed lifetime optimized routing (DLOR) algorithm to conduct the shortest path routing by asynchronous Bellman-Ford algorithm. A general shortcoming of these heuristic algorithms is the lack of theoretical guarantee for reaching the optimal lifetime.

3. Probabilistic routing algorithms: In routing protocols of sensor networks, probabilistic routing was proposed for a sender to choose one of its potential receivers to relay the message based on different kinds of routing probability assignment strategies, such as the geographic routing [25, 26], receiver contention routing and multi-path probabilistic routing [27, 28]. Some studies have revealed the benefits of probabilistic routing. Firstly, due to its natural multi-path property, it can provide better multi-hop reliability than static routing protocols in the presence of link dynamics [27, 28]. Secondly, it is particularly suitable for networks where sensors randomly work at on/off state or in a duty cycle mode. The forwarders contend in real-time in each hop will improves the network energy efficiency and the communication reliability [29, 30]. However, current studies seldom addressed the question of how the distributed probabilistic routing algorithm can maximize the network lifetime.

4. Others: Zhu and Girod [31] proposed an algorithm to minimize congestion by decomposing the total target flow rate into a sequence of rate increments, in which the classical Bellman-Ford algorithm was adopted to find a corresponding minimum-cost route for each increment. Tsai et al. [32] proposed a routing protocol for load balancing based on path energy and self-maintenance. Mauro [33] proposed a gossip-based distributed balancing algorithm for which the nodes need to have extensive information about their neighborhood. Most recently, Lee et al. [34] applied DAG's (Direction Acyclic Graph) to different data collection cycles for load balancing so that energy efficiency was improved in both the time and the spatial domains.

8 Conclusion

In this paper, we have investigated how distributed probabilistic routing can online optimize the remaining lifetime of the network. Sufficient conditions for network lifetime optimization are derived and a DAPR was presented, which has the following characteristics: (1) only based on local information, (2) convergence guaranteed; (3) optimality guaranteed, and (4) fast adaptation to network dynamics. In future research, the performance of DAPR can be improved by joint optimization of routing probability, node mobility or transmission power. MAC protocols for probabilistic routing are also worth of investigating. Further studies can also DAPR without the minimum hop routing constraint. DAPR can also be explored to control forwarder selection in opportunistic routing protocols.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (No. 61202360), the Fundamental Research Funds for the Central Universities in China (No. 21614324), and the Natural Science Foundation of Guangdong Province (No. 2014A030310172).

References

1. Wang, H., Agoulmine, N., Ma, M., & Jin, Y. (2010). Network lifetime optimization in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, *28*(7), 1127–1137.
2. Degirmenci, G., Kharoufeh, J. P., & Prokopyev, O. A. (2014). Maximizing the lifetime of query-based wireless sensor networks. *ACM Transactions on Sensor Networks*, *10*(4), 56:1–56:24.
3. Powell, O., Leone, P., & Rolim, J. (2007). Energy optimal data propagation in wireless sensor networks. *Journal of Parallel and Distributed Computing*, *67*, 302–317.
4. Efthymiou, C., Nikoletseas, S., & Rolim, J. (2006). Energy balanced data propagation in wireless sensor networks. *Wireless Networks*, *12*, 691–707.
5. Jarry, A., Leone, P., Powell, O., & Rolim, J. (2006). An optimal data propagation algorithm for maximizing the lifespan of sensor networks. *DCOSS'2006* (Vol. 4026, pp. 405–421)., Lecture Notes in Computer Science. Berlin/Heidelberg: Springer.
6. Buragohain, C., Agrawal, D., Suri, S. (2005). Power aware routing for sensor databases. *INFOCOM'05*, pages 1747–1757 2881
7. Ghadimi, E., Landsiedel, O., Soldati, P., Duquennoy, S., & Johansson, Mikael. (2014). Opportunistic routing in low duty-cycle wireless sensor networks. *ACM Transactions on Sensor Networks*, *10*(4), 67:1–67:39.
8. Boukerche, A., Efstathiou, D., Nikoletseas, S., & Raptopoulos, C. (2011). Close-to-optimal energy balanced data propagation via limited, local network density information. *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, MSWiM '11* (pp. 85–92). New York: NY, USA, ACM.
9. Dietrich, I., & Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, *5*(1), 5:1–5:39.
10. Liang, W. F., & Liu, Y. Z. (2007). Online data gathering for maximizing network lifetime in sensor networks. *IEEE Transactions on Mobile Computing*, *6*(1), 2–11.
11. Liang, L. W. (2010). Prolonging network lifetime for data gathering in wireless sensor networks. *IEEE Transactions on Computers* (in press).
12. Wu, K., & Liu, A. (1995). Rearrangement inequality. *Mathematics Competitions*, *8*, 53–60.
13. Wang, Y., Wang, Y., Qi, X. (2010). Guided-evolving:convergence to globally optimal load balance by distributed computing using local information, mobicom'10 demo, online at.
14. Yan, T., Bi, Y., Sun, L., & Zhu, H. (2005). Probability based dynamic load-balancing tree algorithm for wireless sensor networks. *Networking and Mobile Computing* (Vol. 3619, pp. 682–691)., Lecture Notes in Computer Science. Berlin/Heidelberg: Springer.
15. Wang, Y., Wang, Y., Qi, X. (2010). Guided-evolution: Convergence to global optimal by distributed computing using local information. *Demo in MobiCom'10, online at:* http://project.iiis.tsinghua.edu.cn/balance.
16. Dag package, http://www-sigproc.eng.cam.ac.uk/atc27/matlab/layout.html.
17. Chang, J.-H., & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, *12*(4), 609–619.
18. Sankar A., Liu, Z. (March 2004). Maximum lifetime routing in wireless ad-hoc networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pages 1089–1097.
19. Dai, H., Han, R. (2003). A node-centric load balancing algorithm for wireless sensor networks. *Globecom'03*, pp. 548–552 4209.
20. Madan, R., & Lall, S. (2006). Distributed algorithms for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *5*(8), 2185–2193.
21. Xue, Y., Cui, Y., Nahrstedt, K. (2005). A utility-based distributed maximum lifetime routing algorithm for wireless networks. In

Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, 2005, pp. 10–18.
22. Jarry, A., Leone, P., & Nikoletseas, S. (2010). Optimal data gathering paths and energy balance mechanisms in wireless networks. *DCOSS'2010* (Vol. 6131, pp. 288–305)., Lecture Notes in Computer Science. Berlin/Heidelberg: Springer.
23. Barrett, C. L., Eidenbenz, S. J., Kroc, L., Marathe, M., & Smith, J. P. (2003). Parametric probabilistic sensor network routing. *WSNA '03* (pp. 122–131). New York: NY, USA, ACM.
24. Chen, Y. R., Yu, L., Dong, Q. F., & Hong, Z. (2011). Distributed lifetime optimized routing algorithm for wireless sensor networks. *Applied Mechanics and Materials*, *40–41*, 448–452.
25. Razzaque, M. A., Hong, C. S. (2008). Load and energy balanced geographic routing for sensor networks. 10th International Conference on Advanced Communication Technology, Vol. I-III, pp. 1419–1422.
26. Nguyen, D. T., Choi, W., Ha, M. T., & Choo, H. (2011). Design and analysis of a multi-candidate selection scheme for greedy routing in wireless sensor networks. *Journal of Network and Computer Applications*, *34*(6), 1805–1817.
27. Huang, X. X., & Fang, Y. G. (2008). Multiconstrained qos multipath routing in wireless sensor networks. *Wireless Networks*, *14*(4), 465–478.
28. Sanati, S., Yaghmaee, M. H., Beheshti, A. (2009). Energy aware multi-path and multi-speed routing protocol in wireless sensor networks. 2009 14th International Computer Conference, pp. 639–644,
29. Sung, E. S., Potkonjak, M. (2009). Localized probabilistic routing for data gathering in wireless ad hoc networks. 2009 7th Annual Communication Networks and Services Research Conference, pp. 356–363.
30. Wu, S. B., & Candan, K. S. (2007). Power-aware single- and multipath geographic routing in sensor networks. *Ad Hoc Networks*, *5*(7), 974–997.
31. Zhu, X. Q., Girod, B. (2005). A distributed algorithm for congestion-minimized multi-path routing over ad hoc networks. *ICME'05*, pp. 1485–1488.
32. Tsai, Y. -P., Liu, R. -S., Luo, J. -T. (2009). Load balance based on path energy and self-maintenance routing protocol in wireless sensor networks. *Lecture Notes in Computer Science*, Vol. 5787, pp. 431–434. Springer: Berlin/Heidelberg
33. Franceschelli, M., Giua, A., & Seatzu, C. (2009). Load balancing over heterogeneous networks with gossip-based algorithms. *American Control Conference (ACC)'09*, pp. 1987–1993.
34. Lee, H., Keshavarzian, A., & Aghajan, H. (2010). Near-lifetime-optimal data collection in wireless sensor networks via spatio–temporal load balancing. *ACM Transactions on Sensor Networks*, *6*, 26:1–26:32.

Yongcai Wang is now an assistant researcher at Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, Beijing, China. He got his B.Sc. degree in 2001 and Ph.D. degree in 2006, both from Tsinghua University. He worked as an associate researcher in NEC Labs China from 2007 to 2009. From 2009 to 2011, he was a post doctor at IIIS. He was a visiting scholar at Cornell University in 2014. His major research interests lie in the broad area of sensor information fusion, network measurement, wireless locating algorithms, and systems. He received the best paper awards at Ubicomm2009 and CWSN2011. He holds several Chinese and international patents on ultrasound locating systems. His research output "ultrasound locating system" was commercialized by NEC and exhibited in World Expo 2014.

Haisheng Tan is currently an associate professor at Jinan University, Guangzhou, China. Before that, he received his BE degree from the University of Science and Technology of China (USTC, with the highest honor) and his Ph.D. degree from the University of Hong Kong in 2004 and 2011, respectively. He worked as a postdoctoral fellow at Institute for Interdisciplinary Information Sciences, Tsinghua University from 2011 to 2013. He was selected into the "Qianbaishi Talent Training Project" at colleges and universities in Guangdong Province in 2014. His major research interests lie in algorithms and wireless networking.