Collaborative Topology Control for Lifetime Maximization

Lei Shi[†] Wen-Zhan Song[†] Mingsen Xu[†] Alex Zelikovsky[†] Li Yu[‡]
[†]Department of Computer Science, Georgia State University, Atlanta, GA 30303
[†]{lshi1@student., wsong@, mxu4@student., alexz@cs.}gsu.edu
[‡]College of Information Engineering, Zhejiang University of Technology, Hangzhou, China
[‡]lyu@zjut.edu.cn

***Abstract*—In a data collection sensor network, how to maximize the network lifetime through topology control remains an open research problem. Previous work has studied this problem by aiming to build a max-lifetime data collection tree, however, tree-based data collection does not necessarily yield maximum network lifetime. In this paper, we consider collaborative multi-path data delivery and formulate the lifetime maximization problem as a max-fair-flow problem, then study how to collaboratively adjust the transmission power of sensor nodes to achieve the max-fair-flow, thus maximizing the network lifetime. We give both theoretical proofs and simulations to validate its correctness and performance.**

***Index Terms*—Collaborative Topology Control; Maximum Lifetime; Energy Efficiency; Wireless Sensor Networks**

I. Introduction

In many existing literatures, the network lifetime is usually defined as the time from the boot of network until the first node in the network dies. For max-lifetime data gathering with or without data aggregation, there has been many research focused on the tree-based protocols. One problem is that the tree-based data collection protocols do not make use of the multi-path for data delivery in the network, thus can not achieve maximum lifetime. We can use an example to illustrate this point.

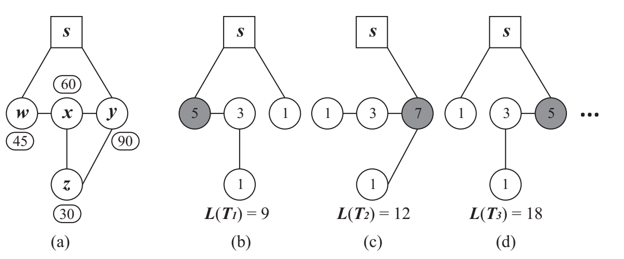

Fig. 1. Network lifetime of single spanning tree. (a) shows the topology of the network, including 4 sensor nodes w, x, y, z and one sink node s. The numbers in ellipses denote the initial battery energy. (b), (c) and (d) are 3 possible spanning trees T_1, T_2 and T_3 induced from the network. Assume T_1, T_2 and T_3 are data collection trees, the number in each node indicates its the energy consumption rate. The grey node is the bottleneck node, the first node to deplete its energy, of each spanning tree. The lifetime L of corresponding spanning trees are 9, 12 and 18 respectively.

A proper example involves a small data collection network with four sensor nodes and one sink node as shown in Fig. 1. The figure also lists three possible spanning trees induced from the data collection network. We assume that each sensor node sends one unit of data per time unit and suppose that transmitting or receiving one data unit consumes one unit of energy. The energy consumption rate of sensor nodes depends on both receiving and transmitting (e.g. in T_1, node w receives 2 and transmits 3 data units per time unit thus consumes 5 units of energy, i.e., the energy consumption rate of node y is 5). Then we can see that the lifetime of listed spanning trees T_1, T_2 and T_3 are 9, 12 and 18 respectively. Note that in tree T_2, the lifetime of bottleneck node y is longer than 12, but after 12 time units the energy left on node y is not enough to transmit one unit of data for every node. In this case, we consider the lifetime of data collection tree as $\lfloor E_B/r \rfloor$ where E_B and r are the energy of bottleneck node and its energy consumption rate. By constructing a spanning tree for data collection in Fig. 1, one can list all the spanning trees induced from the network to get the tree with maximum lifetime. It is not hard to find out that in the example network there is no single spanning tree which can survive more than 18 time units, i.e., no matter which tree is chosen, the network lifetime can not be longer than 18.

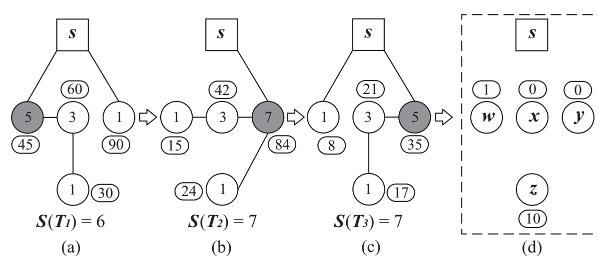

Fig. 2. Network lifetime of scheduling multiple trees. (a), (b) and (c) show the topology of spanning tree T_1, T_2 and T_3 and $S(T_1)$, $S(T_2)$ and $S(T_3)$ indicate the scheduling time of each tree. The numbers in the ellipses are the residual energy before scheduling the tree. (d) shows that after scheduling done, x and y deplete their energy.

Next we will show that the network lifetime can be extended if the sensor nodes make use of the multi-path to transmit data to different parents for data delivery. Instead of employing a single spanning tree as the data collection tree we can use the strategy of scheduling multiple trees. In Fig. 2, by scheduling

978-0-7695-4610-0/11 $26.00 © 2011 IEEE
DOI 10.1109/MSN.2011.26

T_1, T_2 and T_3 for 6, 7 and 7 time units the network can survive 20 time units until x and y die. Scheduling can extend the network lifetime longer than any individual tree. Spanning tree scheduling can be easily formulated as a linear programming problem, and one can find the optimal solution of scheduling by enumerating all different spanning trees (corresponding to the constraints in LP) of a given network. The problem is that the number of spanning trees may be prohibitively large (e.g. exponential in number of nodes for a complete graph). So enumerating all the spanning trees itself is a hard problem with the growth of network size. Berman *et. al.* [1] address the problem of choosing an appropriate subset in the exponential number of constraints to approximate the optimal by applying Garg-Könemann algorithm [2].

Consider the following data collection process. Assuming that there are n sensor nodes and one sink node s in the network, every sensor node tries to produce and deliver one data unit to s in each data collection round. Suppose that after α rounds, the sink node can not collect one data unit from each sensor node in one round. Then the network lifetime is α rounds. Fig. 3 illustrates this process of the example network above. It shows that each node can produce 22 data units then transmit them to sink through the network one by one. After 22 rounds, node w and y can deliver more data to sink but node x and z can not, so the network lifetime is 22 here.

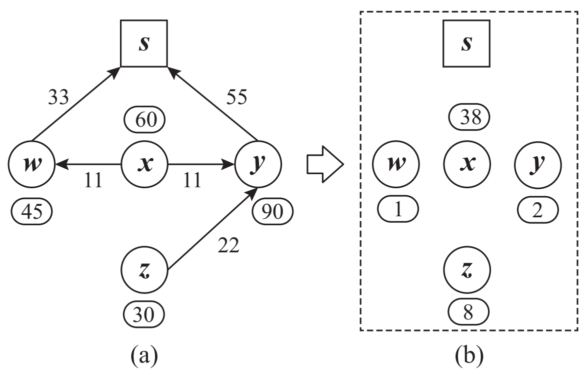

Fig. 3. Network lifetime and max-fair-flow. (a) shows the network topology and initial battery energy of each sensor node, the arrows indicate the data flow direction and the number by each link indicate the data transmitted through it. (b) shows the residual energy of sensor nodes when not all sensor nodes can deliver one data unit to sink in one round.

From the example in Fig. 3 we can see that 22 is actually the data flow that each sensor node can produce and deliver to sink. To maximize the network lifetime we need to maximize this flow amount, i.e., the data collection rounds α.

Definition 1: **Max-fair-flow**. Given a data collection network with n sensor nodes and one sink node s, s can collect one data unit from each node in one *round*. Suppose Given a power assignment of the sensor nodes $\mathcal{P}$, the network satisfies that the minimum dataflow (*rounds*) the nodes archive in the network is maximized and the second lowest dataflow is also maximized and so on. Let a_i be the dataflow of node v_i, then we call $\tilde{\alpha} = \max \alpha_i$ the max-fair-flow of the network.

Notice that, max-fair-flow assignment with topology control also maximizes network lifetime, as explained earlier. However, network lifetime maximization can be achieved through max-min-flow with topology control. Max-fair-flow additionally maximizes throughput and fairness thus a ideal goal of this paper. In this paper, we assume that the node power level is adjustable and the capacity constraint is on nodes, instead of links. Each node capacity is determined by its residual energy, thus it depends on each node's transmission and reception energy, as well as the data rates of sending and receiving. Then we have 1,

Lemma 1: Finding a max-fair-flow $\tilde{\alpha}$ will maximize the lifetime of a data collection network.

1 is straightforward from the previous discussion. We can formulate the network lifetime maximization problem as a max-fair-flow problem, and maximize the flow from the sensor nodes that produce and deliver the minimum amount of data. This is so-called the maxmin fairness flow of the network. We can find such a flow by first maximizing the total flow to sink and then adjusting the flow fairness to guarantee maxmin [3].

We have showed that computing a max-fair-flow can maximize the network lifetime in the network that each sensor node has fixed transmission power. In the existing real-world deployment of sensor networks in data collection, the sensor nodes usually are set with their maximum power level which is not energy efficient. Due to the positive correlation between power level and energy consumption rate of transmission, using maximum power level will impact the network lifetime especially in dense network. The rest of the paper is organized as follows. In section II, we propose an algorithm to collaboratively control topology and maximize the network lifetime by adjusting the transmission power level of sensor nodes. We evaluate the algorithm in simulation and show the results in section III and conclude the paper in section IV.

II. SYSTEM MODEL

Consider a data collection network G with n sensor nodes $(v_1, v_2, ..., v_n)$ in V and one sink node s, all sensor nodes periodically produce data. The sensor data from all the sensor nodes need to be delivered to the sink node. All the sensor nodes are powered by batteries, the initial energy in battery for node v_i is E_i, E_i is finite and the battery is unchargeable, different nodes may have different initial energies. Sink node s is connected to an unlimited power supply where $E_s = +\infty$. Each node can adjust its transmission power from level 0, on which the radio of sensor node is in sleep mode, to level P, on which the node uses maximum transmission power. The energy consumption rate and transmission range depend on the power level, higher power level implies higher energy consumption rate and lager transmission range. Suppose that all the nodes in the network have the same power setting configuration, i.e., each node can set its own power level from 0 to P and the transmission energy consumption rate of different nodes is identical on the same power level, and the energy consumption rate of receiving is fixed at any power level. Besides, we assume that the sensor network is connected, i.e., there exists a path from any node to the sink if all the sensor nodes use their maximum power level.

Let R be the amount of energy required to receive one unit of data and T_p denote the amount of energy required to transmit one unit of data at power level p. p_i denotes the power level of node v_i. Suppose that D_i is the data amount node v_i produces and delivers to sink, M_i is the data amount which can be relayed (received and transmitted) on node v_i. Then for any power level assignment of all sensor nodes $\alpha = \min D_i$ and there exists some power level assignment where α is maximized, i.e., α is the max-fair-flow $\tilde{\alpha}$. Then we can formulate the network lifetime problem as following, **Problem:** Given a network G, find an power level assignment $\mathcal{P} = \{p_i | v_i \in V\}$ such that α is maximized, i.e., the network lifetime is maximized.

To calculate $\tilde{\alpha}$, we use the similar max-flow min-variance approach proposed in [3] which calculates a maximum flow of the network and then balance the flow among different paths. So we need to measure how much data node v_i can transmit at its capacity $C_p(v_i)$ when it uses power level p. Note that as a source in flow network a sensor node needs to first push its own data flow out to maximize the total flow since relaying data for other nodes will consume more energy. So for any node v_i, if $D_i T_{p_i} \geq E_i$ let $C_{p_i}(v_i) = E_i / T_{p_i}$. Otherwise if $D_i T_{p_i} < E_i$ and $D_i T_{p_i} + M_i(R + T_{p_i}) = E_i$ for some M_i, let $C_{p_i}(v_i) = D_i + M_i = D_i + (E_i - D_i T_{p_i})/(R + T_{p_i}) = (E_i + D_i R)/(R + T_{p_i})$. Then given a power level p, node v_i can not transmit more than $C_p(v_i)$ units of data.

III. Algorithm Design and Analysis

Before applying the max-flow min-variance algorithm and adjusting the power level of the sensor nodes, we need to initialize the network with some power level assignment for the sensor nodes. We employ the following scheme, firstly for any sensor node $v_i \in V$, set its power level as maximum. Then if v_i can reach sink, then use the minimum power level that it can connect to sink node, otherwise use the minimum power level that can keep its existing downstream neighbors.

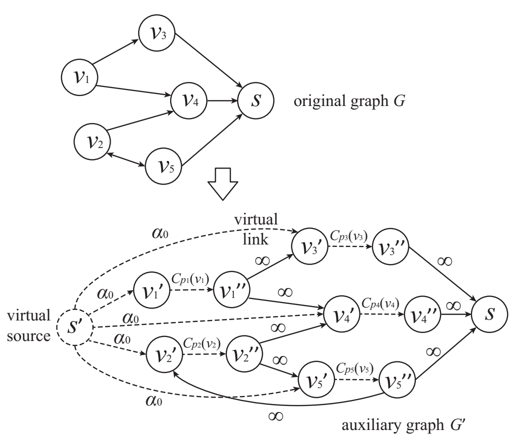

Fig. 4. Construction of auxiliary graph to find the max flow.

This above initialization scheme will keep the network connected and preserve all the residual paths that can be used to generate the max-flow. Choosing such an initialization scheme is based on the following observation. Because all the nodes that can reach sink are hop 1 nodes, and all the flows sent to sink are either sent from or through hop 1 nodes, the total flow amount through the network can not exceed the energy constraint of all hop 1 nodes. Since every hop 1 node uses the minimum power level which allows it to reach sink in this scheme, the node therefore has the maximum capacity with which it can transmit data to sink directly. With this initialization scheme, we can give an upper bound of the network lifetime and the max-flow by upper-bounding $\tilde{\alpha}$.

Let V_h be the set of sensor nodes whose hop-distance to the sink is h, then $V = \sum_{h=1}^{H} V_h$ (where H is the maximum number of hops from sink to any node) and $V_0 = \{s\}$. Let $E_0 = \sum_{v_i \in V_1} E_i$ and q be the minimum transmission power level used by the nodes in V_1, i.e., $q = \min\{p_i | v_i \in V_1\}$. Lemma 2 gives the upper bound of the max-fair-flow $\tilde{\alpha}$.

Lemma 2: Let $\alpha_0 = E_0/(nT_q + (n - |V_1|)R)$, given a network G, $\tilde{\alpha} \leq \alpha_0$.

Proof: Suppose that after initialization, the max-fair-flow of the network is $\tilde{\alpha}$ then we have,

$$\begin{aligned} E_0 &\geq \sum_{v_i \in V_1} \tilde{\alpha} T_{p_i} + \sum_{v_i \in V_1} M_i(R + T_{p_i}) \\ &\geq \sum_{v_i \in V_1} \tilde{\alpha} T_q + \sum_{v_i \in V_1} M_i(R + T_q) \\ &= |V_1| \tilde{\alpha} T_q + (n - |V_1|)\tilde{\alpha}(R + T_q) \\ &= n\tilde{\alpha} T_q + (n - |V_1|)\tilde{\alpha} R \end{aligned}$$

which implies $\tilde{\alpha} \leq E_0/(nT_q + (n - |V_1|)R) = \alpha_0$. ■

Before giving the topology control protocol, we present a modified *Push-Relabel* maximum flow algorithm to be used for calculation of max-flow in the protocol. In the standard *Push-Relabel* max-flow algorithm, the capacity constraints are on the link and the flow is from a single source to a single sink. But in our problem, the capacity constraints are on the nodes and there are multiple sources (all the nodes are sources except the sink node). To facilitate the standard *Push-Relabel* algorithm, we need to transform our problem into the maximum flow problem in the original sense by constructing an auxiliary graph from G (basically expanding V, see Fig. 4).

The auxiliary graph can be constructed as follows. Let G' be the auxiliary graph with node set V' and link set L'. First for each node $v_i \in V$, create two nodes v_i', v_i'' and one link (v_i', v_i'') with capacity $C_{p_i}(v_i)$, add these two nodes and one link to G'. Second, create a virtual source s' and connect s' to each node v_i' with capacity α_0 on link (s', v_i'), add s' and (s', v_i') to G'. At last, for any node $u \in V$, if $(u, v_i) \in L$ (the nodes can reach v_i), create an link (u'', v_i') with capacity ∞; if $(v_i, u) \in L$ (the nodes v_i can reach), create an link (v_i'', u') with capacity ∞. Add (u'', v_i') and (v_i'', u') to G'. Then we get the expanded network G', the capacity constraint on the node is removed and therefore the problem can be treated as

Algorithm 1 Max-Flow Algorithm

Auxiliary graph construction

$G' \leftarrow \emptyset$
$V' \leftarrow V' \cup \{s'\}$
for each node $v_i \in V$ **do**
  $V' \leftarrow V' \cup \{v_i', v_i''\}$
  $L' \leftarrow L' \cup \{(v_i', v_i'')\}$
  $c(v_i', v_i'') = C_{p_i}(v_i)$
  $L' \leftarrow L' \cup \{(s', v_i')\}$
  $c(s', v_i') = \alpha_0$
  for each link $(u, v_i) \in L$ **do**
    $L' \leftarrow L' \cup \{(u, v_i')\}$
    $c(u, v_i') = \infty$
  end for
  for each link $(v_i, u) \in L$ **do**
    $L' \leftarrow L' \cup \{(v_i'', u)\}$
    $c(v_i'', u) = \infty$
  end for
end for

Generic-Push-Relabel algorithm

Initialize-Preflow(G', s')
while there exists an applicable push or relabel operation **do**
  select an applicable push or relabel operation and perform it
end while

a standard maximum flow problem.

Based on the expanded graph G', we apply the *Push-Relabel* maximum flow algorithm on page 674 of [4]. Let $c(u, v)$ be the residual capacity on link $(u, v) \in L'$ the procedure is described in Algorithm 1.

TABLE I
LIST OF NOTATIONS IN ALGORITHM 1 AND 2

$c(u, v)$	the residual capacity of link (u, v)
$C_{p_i}(v_i)$	the capacity of node v_i with power level p_i
p_i	power level of node v_i
P	maximum power level
$S_{p_i}(v_i)$	the node set which v_i can reach with power level p_i
$\tilde{\alpha}$	max-fair-flow
α_0	the upper bound of max-fair-flow
F_0	the upper bound of max flow
E_i	initial energy of node v_i
T_p	transmission energy consumption rate with power level p
R	receiving energy consumption rate
f	max flow of the network

Then we can propose the collaboratively topology control algorithm based on the power level initialization scheme and the upper bound of $\tilde{\alpha}$. The algorithm consists of two phases, power level initialization and power level adjustment, see Algorithm 2. Table I list the notations used in the algorithm. initialization phase, the algorithm first assigns power level to each sensor node in the network with the initialization scheme mentioned above. After that it initializes $\alpha_0 = (\sum_{v_i \in V_1} E_i)/(nT_q + (n - |V_1|)R)$ and $F_0 = n\alpha_0$ (F_0 is the upper bound of max-flow which can be delivered to sink through the network). Then the algorithm assigns α_0 to D_i for each node v_i since this is the upper bound of the max-fair-flow each sensor node can produce and send to sink. According to the definition of the node capacity, the algorithm initializes the capacity of each node v_i as,

$$C_{p_i}(v_i) = \begin{cases} E_i/T_{p_i} & \text{if } \alpha_0 T_{p_i} \geq E_i \\ (E_i + \alpha_0 R)/(R + T_{p_i}) & \text{otherwise} \end{cases}$$

After initialization, the algorithm will adjust the power level of the sensor nodes to collaboratively control the network topology and maximize the network lifetime. It first uses the modified Push-Relabel algorithm similar to [3] and get the max flow f.

Then the algorithm will initialize the flows in network by pushing a pre-flow α_0 from s' to each v_i'' through v_i' without breaking the capacity constraint on link (v_i', v_i''), and find the max flow f. If $|f|$ is equal to F_0, the algorithm terminates and the network lifetime is determined by α_0. If $|f| < F_0$, the algorithm will try to adjust the power level of saturated nodes and increase the max flow. Let $f_m = f$, then the algorithm sorts all the saturated sensor nodes with ascending order of hop distance, puts them in set N and starts to adjust the power level of the nodes with lower hop distance. When adjusting the power level of node v_i, the algorithm decreases the power level of v_i by one, recalculate $C_{p_i}(v_i)$ and the max flow f. If $|f|$ is equal to F_0, the algorithm terminates. If $|f| \geq |f_m|$, $f_m = f$ and the algorithm sorts all saturated nodes again then starts over to adjust the power level of the first node in N. If $|f| < |f_m|$, then the algorithm sets back the power level of node v_i, recalculates $C_{p_i}(v_i)$ and tries to adjust the power level of the next node in N.

After tried all the saturated nodes, if no one can adjust its power level we will use the Min-Variance Flow Balance algorithm from [3], and maximize the network lifetime by balancing the flow in network to get the max-fair-flow $\tilde{\alpha}$. The Min-Variance Flow Balance algorithm iteratively selects the sensor nodes with maximum and minimum source flow (flow data produced and sent from the node). Assume f^u_{max} and f^v_{min} are the max and min source flows where the source node are u and v. If there is an augmenting path between these two nodes, then the algorithm will balance the flow between them. This algorithm repeats this procedure until the minimum flow can not increase and we can get $\tilde{\alpha}$.

Next we discuss the rationale behind this algorithm by the following lemmas and theorem.

Lemma 3: After initialization, for any sensor node $v_i \in V$, increasing its power level can not generate a larger max flow in network G.

Proof: Suppose after initialization the max flow in G is f, then any path from virtual source s' to sink s is saturated. If we increase the power level of node v_i and get a larger max flow, there must exist some augmenting path. If node v_i is saturated, increasing its power level will decrease its capacity and can not yield any augmenting path since v_i is

Algorithm 2 Power Adjustment Algorithm

Power level initialization.

1: **for** each node $v_i \in V$ **do**
2: $\quad p_i \leftarrow P$
3: $\quad$ **if** v_i can reach sink s **then**
4: $\quad\quad$ **while** v_i can reach sink s **do**
5: $\quad\quad\quad p_i \leftarrow p_i - 1$
6: $\quad\quad$ **end while**
7: $\quad\quad p_i \leftarrow p_i + 1$
8: $\quad$ **else**
9: $\quad\quad S_P(v_i) = S_{p_i}(v_i)$
10: $\quad\quad$ **while** $S_{p_i}(v_i) = S_P(v_i)$ **do**
11: $\quad\quad\quad p_i \leftarrow p_i - 1$
12: $\quad\quad\quad$ Update $S_{p_i}(v_i)$
13: $\quad\quad$ **end while**
14: $\quad\quad p_i \leftarrow p_i + 1$
15: $\quad$ **end if**
16: **end for**
17: $\alpha_0 \leftarrow (\sum_{v_i \in V_1} E_i)/(nT_q + (n - |V_1|)R)$
18: $F_0 \leftarrow n\alpha_0$
19: **for** each node $v_i \in V$ **do**
20: $\quad D_i \leftarrow \alpha_0$
21: $\quad$ Initialize $C_{p_i}(v_i)$
22: **end for**

Power level adjustment

1: Find a max flow f by Algorithm 1
2: **if** $|f| = F_0$ **then**
3: $\quad$ TERMINATE
4: **else**
5: $\quad f_m \leftarrow f$
6: $\quad$ Sort all the saturated nodes by ascending order of hop-distance and put them in node set N.
7: $\quad$ **for** $i \leftarrow 1$ to $|N|$ **do**
8: $\quad\quad p_i \leftarrow p_i - 1$
9: $\quad\quad$ Recalculate $C_{p_i}(v_i)$
10: $\quad\quad$ Find a max flow f by Algorithm 1
11: $\quad\quad$ **if** $|f| = F_0$ **then**
12: $\quad\quad\quad$ TERMINATE
13: $\quad\quad$ **else if** $|f| \geq |f_m|$ **then**
14: $\quad\quad\quad f_m \leftarrow f$
15: $\quad\quad\quad$ Go to 6
16: $\quad\quad$ **else**
17: $\quad\quad\quad p_i \leftarrow p_i + 1$
18: $\quad\quad\quad$ Recalculate $C_{p_i}(v_i)$
19: $\quad\quad\quad i \leftarrow i + 1$
20: $\quad\quad$ **end if**
21: $\quad$ **end for**
22: $\quad$ Calculate $\tilde{\alpha}$ by Min-Variance Flow Balance algorithm and TERMINATE
23: **end if**

still saturated. If node v_i is not saturated and $v_i \in V_h (h > 1)$, according to the initialization scheme v_i can not reach more neighbors by increasing power level, so all the paths through v_i are still saturated and no augmenting path exists. If $v_i \in V_1$, increasing its power level may create more paths from v_i to sink s, but any path from s' to v_i is saturated so any path from s' to s is saturated and no augmenting path exists. By contradiction, no larger max flow can be generated. ∎

Theorem 4: In algorithm 2, given a network G, if the max flow is f and $|f| = F_0$ then f is optimal.

Proof: Because each node in the network can generate and deliver at most α_0 data flow to sink according to the flow initialization when we apply the Push-Relabel algorithm. If we find a max flow f where $|f| = F_0$, then each node exactly produces and delivers α_0 data flow to sink since $F_0 = n\alpha_0$. By lemma 2, f is the optimal. ∎

From lemma 3, to increase the network max flow we only need to consider decreasing the power level of sensor nodes after initialization. According to theorem 4, we can also imply that if we find a max flow f where $|f| = F_0$, then the network lifetime is maximum. The proof above shows that if $|f| = F_0$ each sensor node exactly produces and delivers α_0 data flow to sink, and we have showed that α_0 is the upper bound of $\tilde{\alpha}$, therefore the network lifetime is maximum and determined by α_0. Following lemma shows that we can increase the network lifetime by increasing the network max flow.

Lemma 5: In algorithm 2, if we decrease the power level of some sensor node and find a larger max-flow, the max-fair-flow $\tilde{\alpha}$ will not decrease.

Proof: Suppose we decrease the power level of sensor node v_i and find a larger max-flow, then there must be some augmenting paths from virtual source s' to sink s in the network after power level adjustment. According to Min-Variance Flow Balance algorithm, if the node with minimum source flow is on some augmenting path, we can shift more flow to it so $\tilde{\alpha}$ will increase. If the minimum source flow node is not on any augmenting path, then $\tilde{\alpha}$ will not change. ∎

According to lemma 5, the algorithm only needs to iteratively adjust the power level of sensor nodes and increase the network max flow. At the end, if no sensor nodes can be adjusted, the algorithm will balance the flow in the network to increase lifetime. By lemma 1, 3 and 5, we conjecture that after the algorithm done, the power level assignment of the sensor nodes is optimal and maximizes the network lifetime. Theorem 6 gives the worst-case time complexity of the power level adjustment algorithm.

Theorem 6: The execution of power level adjustment in algorithm 2 can terminate within $O(P|V|^4|E|)$ operations.

Proof: In the worst case, the algorithm needs to sort the saturated nodes and run the for loop for $O(P|V|)$ times and each finds $O(|V|)$ max flows by Push-Relabel algorithm. Sorting all the nodes can be done in $O(|V| \log |V|)$ and the Push-Relabel algorithm has $O(|V|^2|E|)$ operations as well as the Min-Variance Flow Balance algorithm. Then the worst-case time complexity of power level adjustment algorithm is $O(P|V|(|V| \log |V| + |V|^2|E||V|) + |V|^2|E|)$. So the algorithm can terminate within $O(P|V|^4|E|)$ operations. ■

IV. Simulation and Performance Analysis

We have evaluated the performance of our algorithm via simulations and finished all the simulations with Java programming. In the simulation, we randomly deployed the sensor nodes in a 100×100 field. The base station is located in $(50, 50)$ where is the center of the field. Each node is randomly assigned an initial energy between 1000 and 10000. Suppose that the power level p for each node can be set from 1 to 10 where the transmission range is set to be $2 \times p$ and the transmission energy consumption rate is set to be $4 \times p$, the receiving energy consumption rate is 10. In order to examine the scalability of the algorithm, we run our algorithm on the network which comprise 40, 60, 80, 100, 120, 140, 160, 180 and 200 nodes respectively.

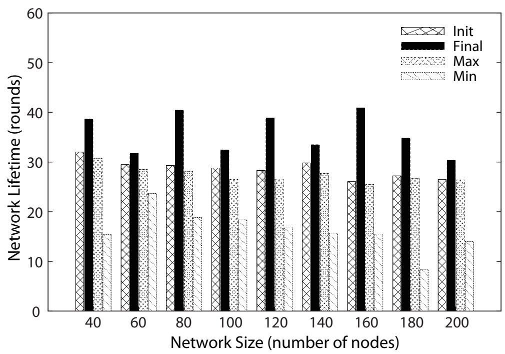

Fig. 5. Network Lifetime

Before running the algorithm, we first check the connectivity for each randomly deployed network since the network lifetime will be zero if some node can not connect to the network. Then with the same network deployment, we evaluate the network lifetime for case (1) the network uses the power assignment derived from the initialization scheme, and (2) the network uses the power assignment calculated by algorithm 2. To illustrate the lifetime performance of the algorithm we also evaluate the network lifetime for case (3) each node uses its max power level for transmitting, and (4) each node uses the same power level p' which is the minimum one to keep the network connected.

The simulation runs 10 times for each network size and we take the average as the result which is shown in Fig. 5. Init, Final, Max and Min denote the network lifetime for case (1), (2), (3) and (4) respectively. We can see that our algorithm initialization already improved the network lifetime compared with case (3) and (4) (e.g. network of size 100, 120 and 140). The power level adjustment fairly increases the network lifetime compared with all other cases. The result varies with network size because we deploy the network and initial energies of sensor nodes randomly. Compared with the best in all 3 other power assignment, the power level adjustment algorithm increases the network lifetime by at least

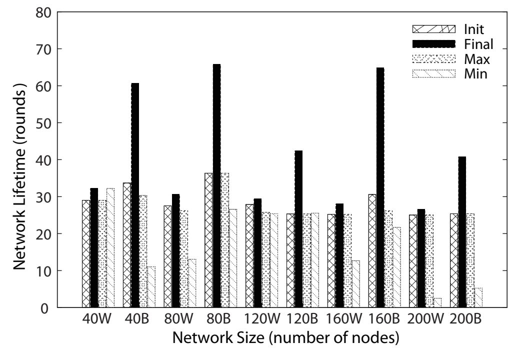

Fig. 6. Best and Worst Cases

7.5%, in network of size 60, up to 57%, in network of size 160.

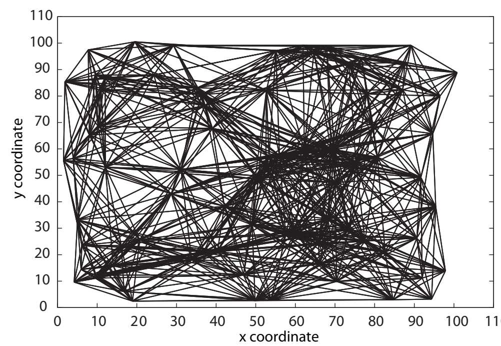

Fig. 7. Network topology with Maximum Power Level

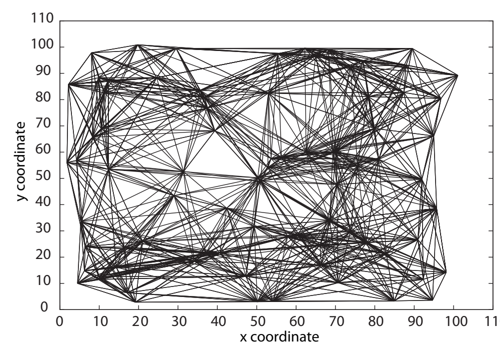

Fig. 8. Network Topology after Initialization Done

As we mentioned above, since the deployment and energy initialization is randomized, the network topology and size may influence the algorithm performance. To further evaluate how bad and how good the performance can be, we run the simulation on selected size of network with different network deployment and sink location for 40 times, then take the best and worst result showing in Fig. 6 where 40W and 40B denote the worst and best result of case (3) (the results from algorithm 2) in all experiments for the network with 40 nodes.

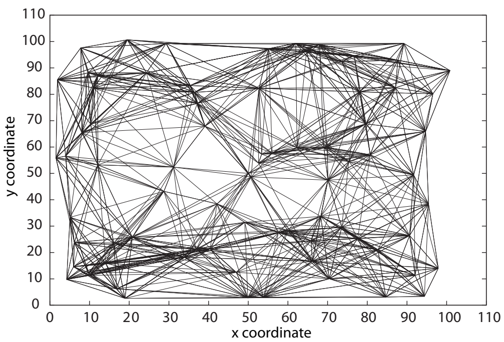

Fig. 9. Network topology after Power Control Done

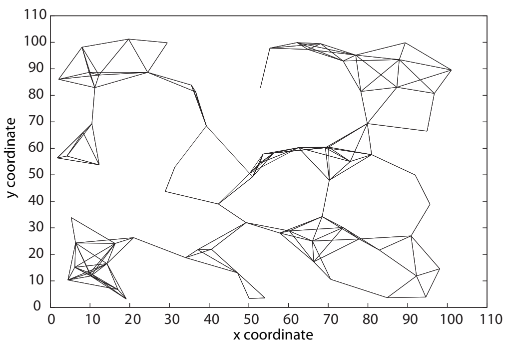

Fig. 10. Network topology with Minimum Power Level

We can see in the worst cases, the algorithm hardly increase the network lifetime for network of size 40 compared with the result of Min. The reason is that in sparse networks the distribution of sensor nodes and sink nodes may cause much fewer chances to send data on multi-path. The algorithm can increase the network lifetime between 5% and 11% for other network size in worst cases. In the best cases, the algorithm increases the network lifetime by at least 60% for network of size 200 and at most 112% for network with 160 nodes.

Next, we use an example to show the density change of the network while applying the power control protocol. Fig. 7 - 10 give the simulation result of the topology change of the network Max, Init, Final and Min case of power assignment mentioned above respectively. In this experiment, we run our algorithm on a network with 80 randomizedly deployed nodes in the area and the sink is at (50, 50), x and y axis in the figures indicate the x and y coordinates of the sensor nodes in the network. The lifetime of the network before and after power control are 26 (Init) and 53 (Final) respectively. Fig. 7 shows the network topology with maximum power level for all nodes. In this topology, there are totally 1871 links and the average degree of the nodes is 23.4. Once the initialization done, the total number of links and the average degree reduce to 1384 and 17.3 8, about 26%. After applying our power control protocol, we can see that the topology in Fig. 9 becomes more sparser, the network has 1143 links which is around 17% less than Init case. Fig. 10 shows the topology of the network where each node uses the same power level which is the minimum one to keep the network connected. The topology of Min case is much more sparser than others, but in this case the network lifetime is only 23, since in this case some nodes may need to transfer more data for other nodes because of the lack of links, and these nodes will deplete their energy faster.

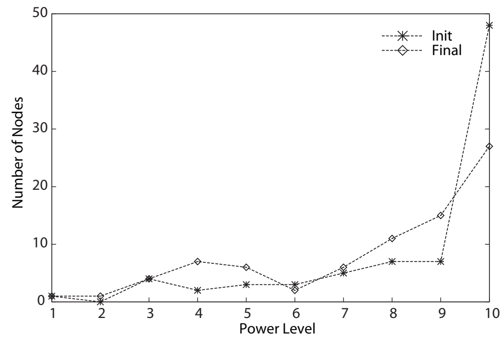

Fig. 11. Network Power Level

In the same test setup above, we also count the nodes with different power levels and show the result in Fig. 11. to see how much energy the network can save with the power control protocol. The x and y axis indicate the power level and the number of nodes with specified power level in the network. From 11, we can see that when initialization done, there 48 nodes which use the maximum power level. After applying the power control protocol, the number of nodes with maximum power level reduces to 27 and the power level distribution in the network is more average. From our setup of parameters for the simulation, in this example, the power control protocol can help the network save about 24.5% of energy in one round of data collection.

V. Related Work

For data collection problem in continuous monitoring applications, several tree-based topology control protocols were proposed in [5], [6], [7], [8], [9], [10] and [11]. Some works focused on tree construction for data gathering. Goel et. al. [12] studied the problem of finding efficient trees to send information from multiple sources to a single sink with data aggregation. They proposed a randomized tree construction algorithm that approximates a tree which is a good approximation simultaneously to the optimum trees for all concave cost functions. In [13] Enachescu et. al. considered a grid of sensors that satisfies the appropriate collision time condition, and proposed a simple randomized tree construction scheme that achieves a constant factor approximation to optimum aggregation tree.

Beside theoretical work, some other research focused on the empirical studies and several practical topology control protocols are proposed. These research are mainly about making the local topology control decisions based on models from the studies of practical experiments. In [14], Son et. al. experimentally investigated the impact of variable transmission power on link quality, and proposed variable power link quality control techniques to enhance the performance of data delivery in wireless sensor networks. [15] addressed the issue that the quality of radio communication between low power sensor devices varies significantly with time and environment. A lightweight algorithm of Adaptive Transmission Power Control for wireless sensor networks was proposed where each node adjust the transmission power based on its neighbors information. Gregory et. al. presented an adaptive and robust topology control protocol in [16]. This protocol uses the packet reception rate instead of received signal strength or link quality indicator to indicate the link quality for robustness. All these empirical work utilized the transmission power adjustment to control topology.

To maximize the network lifetime, several works have been done on the construction of data collection tree. Yan *et. al.* [17] proposed an algorithm which arbitrarily selects a spanning tree of the network and iteratively reduces the load on bottleneck nodes (nodes likely to soon deplete their energy due to high degree or low remaining energy) in order to maximize the network lifetime. Another research work in [18] extended this by exploiting the concept of the bottleneck nodes but without the data aggregation assumption. In these works, the transmission power and range are fixed, i.e., the network physical connection is determined and they assume data is delivered through a tree structure. As shown in the examples of introduction, a tree-based topology may not achieve the optimum. Inspired by the empirical work in topology control, we try to maximize the network lifetime by using the transmission power control and utilizing the multi-path of the sensor network in our research.

In the process of constructing maximum lifetime data collection tree, some other research involve the MDST (Minimum Degree Spanning Tree) problem. Blin et al. [19] focused on the problem of finding a distributed approximated algorithm for a MDST. This research presented the first distributed algorithm on general graphs for the MDST. The algorithm is asynchronous and works for named asynchronous arbitrary networks. [20] proposed a self-stabilizing algorithm for constructing a MDST in undirected networks. Starting from an arbitrary state, the algorithm is guaranteed to converge to a legitimate state describing a spanning tree whose maximum node degree is bounded. This algorithm was the first self-stabilizing solution for the construction of a minimum-degree spanning tree in undirected graphs.

VI. Conclusions

In this paper, we studied the problem of lifetime maximization in data collection network by considering topology control with transmission power adjustment of sensor nodes. A novel transmission power level initialization and adjustment algorithm is proposed to collaboratively control the topology and maximize the network lifetime. Simulation results shows that our algorithm can fairly prolong the network lifetime compared with existing power assignment schemes. Since the Push-Relabel algorithm can be easily implemented as a distributed one, the future work involves proposing a distributed flow balance algorithm and designing a distributed collaborative topology control system to maximize the sensor network lifetime.

References

[1] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power Efficient Monitoring Management in Sensor Networks," in *WCNC*, Mar. 2004.

[2] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *FOCS*, 1998.

[3] M. Xu and W.-Z. Song, "Collaborative Data Delivery in Energy-Synchronized Sensor Networks," in *INSS*, Jun. 2011.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. McGraw-Hill Book Company, 2001.

[5] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Computer Networks*, vol. 42, no. 6, pp. 697–716, Aug. 2003.

[6] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," 2003.

[7] H. O. Tan and I. Körpeoğlu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks," *SIGMOD Record*, vol. 32, no. 3, pp. 66–71, 2003.

[8] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proceedings of the third international symposium on Information processing in sensor networks (IPSN)*, 2004.

[9] C. Hua and T.-s. P. Yum, "Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 892–903, 2008.

[10] M. Khan, G. Pandurangan, and A. Vullikanti, "Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 1, pp. 124–139, 2009.

[11] H. Zhang and H. Shen, "Balancing Energy Consumption to Maximize Network Lifetime in Data-gathering Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1526–1539, 2009.

[12] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk," 2003.

[13] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, "Scale-free aggregation in sensor networks," *Theoretical Computer Science*, 2005.

[14] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," 2004.

[15] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: adaptive transmission power control for wireless sensor networks," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2006, pp. 223–236. [Online]. Available: http://dx.doi.org/10.1145/1182807.1182830

[16] G. Hackmann, O. Chipara, and C. Lu, "Robust Topology Control for Indoor Wireless Sensor Networks," in *Proc. 6th ACM conference on Embedded*, Nov. 2008.

[17] Y. Wu, S. Fahmy, and N. B. Shroff, "On the Construction of a Maximum-Lifetime Data Gathering Tree in Sensor Networks: NP-Completeness and Approximation Algorithm," in *INFOCOM*, Apr. 2008.

[18] J. Liang, J. Wang, J. Cao, J. Chen, and M. Lu, "An Efficient Algorithm for Constructing Maximum lifetime Tree for Data Gathering Without Aggregation in Wireless Sensor Networks," in *Mini-Conference of IEEE INFOCOM*, 2010.

[19] L. Blin and F. Butelle, "The First Approximated Distributed Algorithm for the Minimum Degree Spanning Tree Problem on General Graphs," in *International Parallel and Distributed Processing Symposium (IPDPS)*, Apr. 2003.

[20] L. Blin, M. G. Potop-Butucaru, and S. Rovedakis, "Self-stabilizing minimum-degree spanning tree within one from the optimal degree," in *International Parallel and Distributed Processing Symposium (IPDPS)*, May 2009.