Distributed Progressive Algorithm for Maximizing Lifetime Vector in Wireless Sensor Networks

Liang Zhang† Shigang Chen† Ying Jian† Yuguang Fang‡
†Department of Computer & Information Science & Engineering, University of Florida
‡Department of Electrical and Computer Engineering, University of Florida

Outline
- Motivation
- Related Work and Existing Problem
- Problem Description
- Solution
- Simulation
  - Simulation Setup
  - Convergence Speed of DPA
  - Scalability of DPA
  - Comparison with Hou’s Centralized Algorithm
  - Comparison with Other Centralized and Distributed Solutions
- Conclusion

Motivation (1)
- What is exactly the lifetime of a sensor network?
- Many prior works [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] define the network’s lifetime as the time before the first sensor in the network runs out of energy, or before the first loss of coverage [11].

Motivation (2)
- However, when one sensor dies, the rest of the network can still work, as long as useful data generated by other sensors can still reach the sink.

Motivation (3)
- An appropriate definition for the lifetime of a sensor network should include the lifetimes of all sensors that produce useful data.
  - A sensor’s lifetime is the duration from the time when it begins to generate the first data packet to the time when it generates the last packet that is deliverable to the sink.
  - The network’s lifetime can be defined as the vector of all sensors’ lifetimes sorted in ascending order, which is called the lifetime vector.

Motivation (4)
- The value of the lifetime vector is determined by the nodes’ packet forwarding policies that specify how packets are forwarded from the sensors through the network to the sink.
Related Work (1)

- Hou et al. [12], [13] define the problem of maximizing a sensor network’s lifetime as to find the packet forwarding policies for all nodes that collectively produce the lexicographically largest lifetime vector, called the maximum lifetime vector.

Related Work (2)

- Hou et al. show that this problem can be modeled as a series of linear programming (LP) problems.
- After solving the LP problems, the sink uploads the optimal packet forwarding policies to the sensors. Based on its forwarding policy, each sensor forward its packets. Such a solution is however a centralized one.

Existing Problem (3)

- Collecting the complete information about the network and uploading the complete forwarding policies to all nodes require significant amount of transmissions in the network, particularly for nodes around the sink.
- To avoid these problems, a distributed algorithm that spreads the overhead evenly on all nodes becomes important.

Concepts-Routing Graph

Let \( D_i \) be the set of neighbors that node \( i \) use as the next hop to the sink. They are called downstream neighbors of node \( i \).

\[
D_7 = \{4, 3\}
\]

(7, 4) and (7, 3) are called outgoing links of node 7.

Let \( U_i \) be the set of upstream neighbors, which use \( i \) as the next hop on their routing paths to the sink.

\[
U_7 = \{11\}
\]

(11, 7) is called incoming link of node 7.

Let \( N \) be the set of sensor nodes,

\[
E = \{(i, j) \mid \forall i \in N, j \in D_i\}
\]

We call the graph consisting of all these links as the routing graph.

Concepts-volume and source volume

- The volume \( v(i, j) \) of a link \((i, j)\) is defined as the number of packets transmitted on the link over the lifetime of the sensor network.
- The source volume \( v(i) \) of a node \( i \) is defined as the number of new data packets generated by \( i \).
Concept-feasible volume schedule

Let \( e_i \) be the energy available at node \( i \). Let \( \alpha \) be the amount of energy that a node spends on receiving a data packet from an upstream neighbor, \( \beta \) be the amount of energy that node \( i \) spends on producing a new data packet, \( \gamma_i \) be the amount of energy that node \( i \) spends on sending a packet. The energy constraint is given below.

\[
\sum_{k \in U_i} \alpha \times v(k, i) + \beta_i \times v(i) + \sum_{j \in D_i} \gamma_i \times v(i, j) \leq e_i, \quad \forall i \in N
\]

(1)

We say a node \( i \) is exhausted if

\[
\sum_{k \in U_i} \alpha \times v(k, i) + \beta_i \times v(i) + \sum_{j \in D_i} \gamma_i \times v(i, j) = e_i.
\]

Concept-lifetime and lifetime vector

- For an arbitrary feasible volume schedule, we can calculate the lifetime of each data source \( s \in S \) as follows:

\[
t_s = v(s)/g_s
\]

Where \( g_s, s \in S \), be the source rate at which node \( i \) generates new data packets.

The lifetime vector of the sensor network is defined as \((t_s, s \in S)\) sorted in ascending order.

Concept-lifetime vector

- One lifetime vector \( T1 \) is greater than another \( T2 \) if \( T1 \) is lexicographically larger than \( T2 \).
- For example,

\[
T1 = (2, 3, 3, 4) \succ T2 = (2, 3, 3, 3)
\]

Problem Description

- The maximum lifetime vector problem is to find a feasible volume schedule that produces the largest (or say, maximum) lifetime vector.

Solution

- Distributed progressive algorithm (DPA) consists of
  - Initialization phase
  - Iterative phase
    - Step 1: From Rates to Volume Bounds
    - Step 2: From Volume Bounds to Volumes and Rates

DPA - Initialization phase

- \( r(5, 1) = g_5/1 \);
- \( r(6, 4) = g_6/1 \);
- \( r(7, 4) = r(7, 3) = g_7/2 \);
- \( r(8, 3) = g_8/1 \);

If \( k \) has no upstream neighbor (i.e., \( k \) is a leaf in the routing graph), it distributes its source rate evenly among its outgoing links, i.e., \( r(k, i) = \frac{g_k}{|D_k|}, \forall i \in D_k \), where \( r(k, i) \) is rate value to link \((k, i)\).
DPA - Initialization phase

\[ r(4,1) = (r(6,4)+r(7,4)+g_4)/2 \]
\[ r(4,2) = r(4,1) \]

After a node \( i \) learns \( r(k,i) \) in RATE packets from all upstream neighbors \( k \), it first computes its outgoing rates as \( r(i,j) = \sum_{k \in D_i} r(k,i) \), \( \forall j \subseteq D_i \), and then sends those rates to downstream neighbors \( j \) in a RATE packet.

Return

DPA - iterative phase

-Step 1: From Rates to Volume Bounds

\[ b(1,\text{sink}) = \infty; \quad b(2,\text{sink}) = \infty \]
\[ b(4,2) = (b(2)/g_2) \cdot r(4,2) \]
\[ b(3,2) = (b(2)/g_2) \cdot r(3,2) \]
\[ b(4,2) + b(3,2) + b(2) \leq b(2, \text{sink}) = \infty \]
\[ b(2, \text{sink}) = b(4,2) + b(3,2) + b(2) \leq \infty \]
\[ \alpha(b(4,2) + b(3,2)) + \beta b(2) + \gamma b(2, \text{sink}) \leq \infty \]

Return

DPA - iterative phase

-Step 1: From Rates to Volume Bounds

\[ b(1,\text{sink}) = \infty; \quad b(2,\text{sink}) = \infty \]
\[ b(k,i,j) = \frac{b(k,j)}{g_j} \cdot r(k,i) \]
\( \forall k,j \in (1 \ldots |D|), r(k,i) \geq 0, g_j > 0, b(k,j,i) \neq 0 \)

\[ \sum_{k \in D_i} b(k,i) + b(i) \leq \sum_{j \in D_i} b(i,j) \]

Return

DPA - iterative phase

-Step 2: From Volume Bounds to Volumes and Rates

Vol_Rate Packet

When the sink receives Vol_Rate from all upstream neighbors, it knows that Step 2 is completed.

Figure 2: Iterations of DPA

DPA - Termination Conditions

- A node sets the flag if it changes a link volume by an amount that is not negligibly small.
- It is up to the application requirement to decide on how small is negligible.
- DPA may also be terminated artificially after a certain number of iterations,
- or when the resulting lifetime vector meets the application requirement.
Simulation

- Convergence Speed of DPA
- Scalability of DPA
- Comparison with Hou’s Centralized Algorithm
- Comparison with Other Centralized and Distributed Solutions

Convergence Speed of DPA

- Consider the lifetime vector $V_i$ produced by DPA after the $x$-th iteration.
- Let $t_i(s)$ be the lifetime of source $s$ in $V_i$. Let $t_i(s)$ be the lifetime of $s$ in MLV.

$$\max_{s \in S} \left( \frac{t_i(s) - t_i(s)}{t_i(s)} \right), \quad \frac{1}{|S|} \sum_{s \in S} \left( t_i(s) - t_i(s) \right).$$

Scalability of DPA

Comparison with Hou’s Centralized Algorithm

Comparison with Other Centralized and Distributed Solutions

- SLP (following the same name used in [12]) that is a linear programming solution for maximizing the minimum lifetime of all sources, and
- MPR (Minimum-Power Routing [15], [16]) that is a distributed algorithm for energy-efficient routing.

Fig. 7: Left plot: comparison of nodal overhead distribution between LP and DPA. Right plot: comparison of maximum nodal overhead between LP and DPA.

Fig. 6: Comparison of running time between LP and DPA.
Conclusion

- We have proposed a distributed progressive algorithm for maximizing the lifetime vector in a wireless sensor network, the first algorithm of its kind for this problem.
- The design of the algorithm was based on the necessary and sufficient conditions that we have proved for producing the maximum lifetime vector.
- Simulations are performed to demonstrate the performance of the algorithm.

Thank you so much!

Q&A