Lecture 3: Energy-aware Routing Algorithms

Sotiris Nikoletseas
Associate Professor

CEID - ETY Course
2015 - 2016
Overview

A. The energy balance problem
B. A distance-based probabilistic energy balance protocol
C. Another energy balance problem (residual-energy based)
D. Other methods
  - Adjusting transmission ranges
  - An offline approach
  - Power aware routing
  - Lifetime maximizing routing
  - Load-balanced energy-aware routing
A. The Energy Balance Problem (I)

Representative data propagation protocols

- Directed Diffusion (DD): a *tree-structure* protocol (suitable for low dynamics)
- LEACH: *clustering* (suitable for small area networks)
- Local Target Protocol (LTP): *greedy, single-path optimization* (best for dense networks)
- Probabilistic Forwarding Protocol (PFR): *redundant optimized* transmissions (good efficiency / fault-tolerance trade-offs, best for sparse networks)
- Energy Balance Protocol (EBP): guaranteeing *same per sensor energy* (prolong network life-time)
The Energy Balance Problem (II)

All protocols tend to “strain” some specific nodes in the network.

- In a *hop-by-hop* scheme the nodes *closer to the sink* tend to be overused.
- In a *direct transmission* scheme the *distant nodes* tend to be overused.

“How can we achieve *equal energy dissipation per node* in order to prolong the network lifetime by avoiding early network disconnection?”
B. A Probabilistic Energy Balance Protocol (distance-based)

- Direct transmission cost is much larger than that of one-hop transmission and exhausts distant nodes.
- One-hop transmissions are cheap but tend to overuse nodes that lie closer to the sink.

Solution: “Each node chooses randomly whether to propagate the event one-hop closer to the sink or to transmit it directly to the sink”.

The goal is to “balance” the two types of transmissions and achieve equal average energy dissipation per sensor.
**Network Partition:**

- *Partition* the network into *n* sectors, "slices", of width *R* (the transmission range).
• **Data Propagation:**
  
  Each node in sector \( i \) propagates its messages according to the following rule:
  
  - Propagate the message to sector \( i - 1 \) with probability \( p_i \).
  - Propagate the message *directly to the sink* with probability \( 1 - p_i \).

  The *choice* of \( p_i \) is made such as the *average per sensor energy dissipation is the same* for all sensors in the network.
An Instance of the Execution

There is a message on node $i$. 
**Balanced Average per Sensor Energy Dissipation**

- $S_i$: the *area size* of sector $i$.
- $\mathcal{E}_i$: the *total energy* dissipated in sector $i$.
- Sensor nodes are spread *random uniformly* in the network area.

We want:

$$\frac{E[\mathcal{E}_i]}{S_i} = \frac{E[\mathcal{E}_j]}{S_j} \quad \forall i, j \in \{1, \ldots, n\}$$
The events occur in random uniform positions in the network.

Let $\epsilon_{ij}$ a *random variable* that measures the *energy that dissipates* the sector $i$ so as to handle message $j$.

$$
\epsilon_{ij} = \begin{cases} 
  cR^2 & \text{with probability } p_i \\
  c(iR)^2 & \text{with probability } 1 - p_i
\end{cases}
$$

Clearly,

$$
E[\epsilon_{ij}] = [i^2 - p_i(i^2 - 1)]cR^2
$$
Computation of $p_i$ (message handling)

- Let $h_i$ the number of messages that handles sector $i$.
- Let $f_i$ the number of messages that were forwarded to sector $i$.
- Let $g_i$ the number of messages that were generated in sector $i$.

Clearly:

$$h_i = f_i + g_i$$

By linearity of expectation:

$$E[h_i] = E[f_i] + E[g_i]$$
Computation of $p_i$

Lemma

The following relationship holds:

$$ E[f_i] = p_{i+1} E[h_{i+1}] = p_{i+1} \cdot (E[f_{i+1}] + E[g_{i+1}]) $$

Clearly:

$$ p_{i+1} = \frac{E[f_i]}{E[f_{i+1}] + E[g_{i+1}]} $$

The Energy Balance Property:

$$ E \left[ \frac{\sum_{k=1}^{h_i} \epsilon_{ik}}{S_i} \right] = E \left[ \frac{\sum_{k=1}^{h_j} \epsilon_{jk}}{S_j} \right] \quad \forall i, j \in \{1, \ldots, n\} $$
A recurrence relation for $p_i$

$$a(i + 1)E[f_{i+1}] - (d(i) + a(i))E[f_i] + d(i - 1)E[f_{i-1}] = a(i)E[g_i] - a(i + 1)E[g_{i+1}]$$

where

$$a(i) = \frac{i^2}{2i - 1} \quad d(i) = \frac{(i + 1)^2 - 1}{2i + 1}$$

Initial conditions:

$$E[f_n] = 0 \quad E[f_0] = n$$
Computation of $p_i$

Structure of the rest of solution evaluation.

- First we transform the recurrence into a simpler recurrence with only two successive terms (whose coefficient is 1).
- Having solved the latter recurrence, we solve the initial one.
- Having computed the values of $E[f_i]$ for $i = 1, \ldots, n$, we compute the exact values of probabilities $p_i$. 
The exact solution

- The solution:

$$E[f_i] = -\sum_{k=1}^{n-i} \frac{\prod_{j=k}^{n-i+1} a(n-j)}{\prod_{j=k}^{n-i} d(n-j)} \cdot \left( \sum_{j=1}^{n-k} (a(j)E[g_j] - a(j+1)E[g_{j+1}]) + a(1) \cdot E[f_1] \right)$$

where

$$\prod_{i=1}^{i-1} a(i) = 1$$

- Easily computed in a repetitive manner

- Thus we can calculate $p_i$’s:

$$p_{i+1} = \frac{E[f_i]}{E[f_{i+1}] + E[g_{i+1}]}$$
A closed approximate form for $p_i$

If $E[f_i] \approx E[f_{i-1}]$ then

$$p_i = 1 - \frac{3x}{(i + 1)(i - 1)} \quad \text{for } 3 \leq i \leq n$$

$p_2 = x$ and can be set to $\frac{1}{2}$

$E[f_i] \approx E[f_{i-1}]$ is realistic because:

- $S_i \approx S_{i-1}$
- $p_i \approx p_{i-1}$. 

Comments on $p_i$

- when $i$ is large, $p_i$ is large, i.e. when far away from the sink it is better to move hop-by-hop, to avoid spending too much energy.
- when $i$ becomes small, $p_i$ is small, i.e. when we approach the sink it is better to transmit directly in order to bypass the critical region (and since energy consumption is small).
C. Another Energy Balance Protocol
(based on residual energy)

Our algorithm balances energy consumption:

- Slicing of the network
- Generalized data propagation algorithm, allows to jump over bottle-neck nodes

Note: do hops to “intermediate” slices help?
Lifespan Maximization

Main Theoretical Result
Lifespan is maximized by a \textit{mixed} data propagation algorithm

Application
We use this fact to propose a distributed optimal data propagation algorithm

- mixed propagation: only single hops and direct to sink transmissions
- mixed strategies beat every other possible strategy (wrt. lifespan)
**Energy cost:**
Sending a message from slice $i$ to $j$ costs $(i - j)^2E/\text{msg}$

<table>
<thead>
<tr>
<th></th>
<th>sink</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sink</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>9</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
1. **Energy cost:**
   Sending a message from slice $i$ to $j$ costs $(i - j)^2 E/\text{msg}$

2. $f_{i,j}$ is the *message rate* from slice $i$ slice to slice $j \text{ msg/t}$

<table>
<thead>
<tr>
<th></th>
<th>sink</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sink</td>
<td>$f_{0,0}$</td>
<td>$f_{0,1}$</td>
<td>$f_{0,2}$</td>
<td>$f_{0,3}$</td>
<td>$f_{0,4}$</td>
</tr>
<tr>
<td>1</td>
<td>$f_{1,0}$</td>
<td>$f_{1,1}$</td>
<td>$f_{1,2}$</td>
<td>$f_{1,3}$</td>
<td>$f_{1,4}$</td>
</tr>
<tr>
<td>2</td>
<td>$f_{2,0}$</td>
<td>$f_{2,1}$</td>
<td>$f_{2,2}$</td>
<td>$f_{2,3}$</td>
<td>$f_{2,4}$</td>
</tr>
<tr>
<td>3</td>
<td>$f_{3,0}$</td>
<td>$f_{3,1}$</td>
<td>$f_{3,2}$</td>
<td>$f_{3,3}$</td>
<td>$f_{3,4}$</td>
</tr>
<tr>
<td>4</td>
<td>$f_{4,0}$</td>
<td>$f_{4,1}$</td>
<td>$f_{4,2}$</td>
<td>$f_{4,3}$</td>
<td>$f_{4,4}$</td>
</tr>
</tbody>
</table>
Generalized-flow maximization

Problem

Given a WSN: Maximize the generalized network flow

- Given the detection rates $f_{0,i} := g_i$
- Given the available energy: $b_i$

LP to maximize the generalized-flow:

Maximize $T$ in:

1. $f_{0,i} = Tg_i$ Detection rates
2. $\sum_{j=0}^{N} f_{i,j}(i - j)^2 \leq b_i$ Energy constraint
3. $\sum_{j=0}^{N} f_{i,j} = \sum_{j=0}^{N} f_{j,i}$ Flow equations

- $g_i$: event generation rates (i.e. data injected at $i$)
- maximize time $T \Rightarrow$ lifespan maximization
Mixed-flow maximization

**Problem**

*Given a WSN: Maximize the mixed network flow*

- Given the detection rates $f_{0,i} := g_i$
- Given the available energy: $b_i$

**LP to maximize the mixed-flow:**

Maximize $T$ in:

1. $f_{0,i} = Tg_i$  *Detection rates*
2. $\sum_{j=0}^{N} f_{i,j} (i - j)^2 \leq b_i$  *Energy constraint*
3. $\sum_{j=0}^{N} f_{i,j} = \sum_{j=0}^{N} f_{j,i}$  *Flow equations*
4. $f_{i,j} = 0$ if $j \not\in \{0, i - 1\}$  *mixed flow constraint*
Mixed-flow maximization

Problem

*Given a WSN: Maximize the *mixed* network flow*

- Given the detection rates $f_{0,i} := g_i$
- Given the available energy: $b_i$

LP to maximize the mixed-flow:

Maximize $T$ in:

1. $f_{0,i} = Tg_i$ Detection rates
2. $\sum_{j=0}^{N} f_{i,j} (i - j)^2 \leq b_i$ Energy constraint
3. $\sum_{j=0}^{N} f_{i,j} = \sum_{j=0}^{N} f_{j,i}$ Flow equations
4. $f_{i,j} = 0$ if $j \not\in \{0, i - 1\}$ mixed flow constraint

(4) guarantess no hops to “intermediate” slices
(3) guarantees flow preservation
## Mixed-flows are optimal

### Result

If there exists an NRG-balanced mixed flow

1. it maximizes the *mixed-flow*
2. it maximizes the *generalized-flow*

### Application

We can maximize the generalized-flow problem using:

- a mixed-flow
- a local property (energy-balance)
A distributed algorithm

### Definition

#### Inputs

1. Each node has a potential $\text{potential}(n) \sim E\text{nergySpent}(n)$
2. Each node knows its list of neighbours $\mathcal{V}_n$
3. Each node knows the potential of its neighbours

#### Algorithm Propagate Data

- Find $m$: the lowest potential neighbour
- If $\text{potential}(m) < \text{potential(self)}$ then send data to $m$
- Else send data directly to the sink
A distributed algorithm

Illustration

Remark

- The algorithm produces a mixed flow
- The algorithm balances energy
A distributed algorithm

Illustration

Remark

- The algorithm produces a mixed flow
- The algorithm balances energy
Stability of the Algorithm
A Markov chain approach

Inputs

- Let $\lambda_i$ be the probability that the event occurs in slice $i$
- $X_i(t)$ is the energy spent by slice $i$ at time $t = 1, 2, 3, \ldots$

We consider

$$X(t) = \begin{pmatrix}
X_N(t) - X_{N-1}(t) \\
X_{N-1}(t) - X_{N-2}(t) \\
\vdots \\
X_2(t) - X_1(t)
\end{pmatrix}$$

Remarks

- $\{X(t)\}_{t \geq 0}$ is a Markov Chain
- If $\lambda_i > 0$, the Markov chain is irreducible

We would like to have $X(t) \xrightarrow{t} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$
Stability of the algorithm

Note: $\lambda_i > 0$ is necessary for irreducibility but not always realistic.

Thus, we give sufficient (martingale type) conditions for stability.

Stability: the MC can leave the origin but will always come back.

Theorem

If $i^2 \lambda_i > (i - 1)^2 \lambda_{i-1}$, the Markov chain $X(t)$ is stable around $\mathcal{A}$, with

$$\mathcal{A} = \left\{ X \in \mathbb{R}^{N-1} : |X_i(t) - X_{i-1}(t)| \leq \frac{i^2}{2}, \; i = 2, \ldots, N \right\}$$
Simulations

- Scatter nodes randomly over a region
- Events are generated randomly
- Data is propagated according to the algorithm
First Simulation

- 1000 sensors randomly dispersed over a 10m disc
- Sink at the center of the disc
- Potential function $potential(n) = Energy(n)$
First Simulation

Network flow

\[ \max\{\text{nodes}\}(\text{EnergySpent}) \]

- \( F \): the flow of the algorithm
- \( U \): maximum possible flow (offline, computed by an LP)
- \( L \): maximum possible flow without direct transmissions
First Simulation

Remarks
- Close to optimal
- Distributed and on-line!
First Simulation

- off-line ideal flow $U$ balances the energy load
- the online algorithm performs very well
Second Simulation

- 600 sensors randomly dispersed over 2 intersecting 30° sector graphs of 10m diameter with one sink at the narrow end of each sector
- events can be reported to either sink
Second Simulation

- 600 sensors randomly dispersed over 2 intersecting 30° sector graphs of 10m diameter with one sink at the narrow end of each sector
- events can be reported to either sink
1. Prove that optimal solution belongs to a subset of realistic data propagation algorithms: a *mixed strategy* which is *energy balanced*

2. Propose a distributed algorithm based on theoretical results

3. Show that the algorithm is efficient (Markov chain context and simulations)
D. Other Methods

- Adjusting transmission ranges
- An offline approach
- Power aware routing
- Lifetime maximizing routing
- Load-balanced energy-aware routing
D1. Adjusting transmission ranges to avoid energy holes

Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting
S. Olariu, I. Stojmenovic
in INFOCOM 2006

- Uniformly distributed sensors, each sending roughly the same number of reports to the sink
- They prove that to minimize energy spent, all sectors must have the same width (which they evaluate)
- This choice, however, leads to uneven energy depletion. Towards energy balance, sector widths must be fine-tuned
- As expected, the width of sectors in energy-balanced network decreases as we near the sink
D2. An offline approach

On the Energy Hole Problem of Nonuniform Node Distribution in Wireless Sensor Networks

X. Wu, G. Chen, S. K. Das
in MASS 2006

- They find that in a circular sensor network with a nonuniform node distribution and constant data reporting, the unbalanced energy depletion among the nodes in the whole network is unavoidable.
- A suboptimal energy efficiency among the inner parts of the network is possible if the number of nodes increases with geometric proportion from the outer parts to the inner ones.
- They also present a routing algorithm with this node distribution strategy.
D3. Power-Aware Routing

- In its basic version, the method selects routes in such a way as to prefer nodes with longer remaining battery lifetime (residual energy).

- Let $R_i$ the remaining energy of an intermediate node $i$. The following link metric is used for all links out of node $i$:

$$C_{ij} = \frac{1}{R_i}$$

- A shortest-cost path algorithm (such as Dijkstra’s or Bellman-Ford) is used to determine a path $\mathcal{P}$, minimizing

$$\sum_{i \in \mathcal{P}} \frac{1}{R_i}$$

This way, nodes with residual energy $R_i$ are favored.
The previous method (avoiding low-energy nodes) avoids early node failures. On the other hand, methods minimizing per-hop transmission costs minimize the total energy spent. To optimize the system lifetime globally, both goals should be addressed simultaneously.

A possible balance is to select the minimum energy path at the beginning (when all nodes have high energy) and avoiding the low residual energy nodes later during the protocol evolution.
This can be implemented by the following link metric:

\[ C_{i,j} = T_{i,j}^a \cdot R_i^{-b} \cdot E_i^c \]

where \( T_{i,j} \) the transmission cost on the \((i,j)\) link, \( R_i \) the residual energy of node \( i \) and \( E_i \) the initial energy of node \( i \).

The choice of the \( a, b, c \) parameters allows addressing different performance priorities and their combinations:

- If \((a, b, c) = (0, 0, 0)\) we get a minimum number of hops protocol.
- If \((a, b, c) = (1, 0, 0)\) we get the minimum total energy per packet protocol.
- If \(b = c\), we have normalized residual energies (e.g. based on initial energy at the node), while \(c = 0\) implies absolute residual energies.
- If \((a, b, c) = (0, 1, 0)\) we get the power-aware routing method described previously.

Simulation results suggest that a non-zero \( a \) and a relatively large \( b = c \) terms (e.g. \((1, 50, 50)\)) provide best performance.
D5. Load-Balanced Energy-Aware Routing (I)

- An *additional level of load balancing* probabilistically spreads the (single) paths out of a forwarding node.
- Let the cost to destination from node $i$ through a candidate neighbor $j$ be given as:

$$C_{ij} = C_j + c_{ij}$$

where $C_j$ is the expected minimum cost-to-destination from $j$ and $c_{ij}$ is any link cost metric (e.g. the $T_{i,j}^a \cdot R_i^{-b} \cdot E_i^c$ metric discussed above). For each neighbor $j$ in its set of candidate neighbors $N_i$, node $i$ assigns a forwarding probability that is inversely proportional to the cost to destination:

$$P_{ij} = \frac{C_{ij}^{-1}}{\sum_{k \in N_i} C_{ik}^{-1}}$$

(i.e., neighbors with lower cost-to-destination are probabilistically favored).
The expected minimum cost to destination for node $i$ is obtained as follows:

$$C_i = \sum_{j \in N_i} P_{ij} C_{ij}$$

Each time the node $i$ needs to route a new packet, it forwards it to any of its neighbor $j$ randomly with the corresponding probability $P_{i,j}$.

This way a potentially different (single) path is chosen for each new packet, preventing the use of the same single path always (and the rapid energy draining which would emerge).
References (I)


