CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATION ENGINEERING

B2M32BTSA - Bezdrátové technologie
BE2M32BTSA - Wireless Technologies and Sensor Networks

Routing in wireless sensor networks

Pavel Mach
Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Telecommunication Engineering
6G mobile RESEARCH LAB

WSN principles

Wireless Sensor Network (WSN)

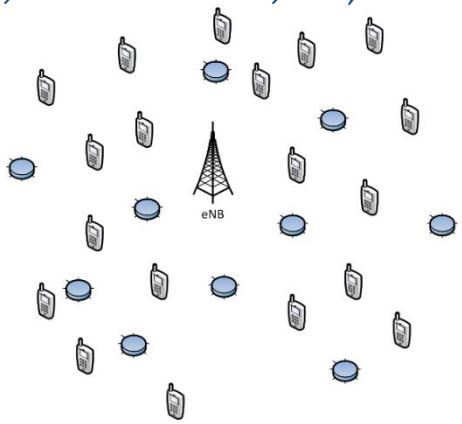
- Set of nodes (sensors) distributed in a field and exchanging information
- Deployed in structured or unstructured manner (with changing topology)

Each Node (Sensor) can act as

- **Source** - Generates data to be sent
- Sink / **Destination** - Receives data generated by Source
- **Relay** of data (Source to Sink)

Challenges and issues for WSN routing

- Dynamic time varying wireless channel and topology (movement, fading, ...)
- Asymmetric wireless channels (multipath propagation, interference, ...)
- Limited resources for route management (signaling)
- Energy constraint for nodes



Flooding

Principle

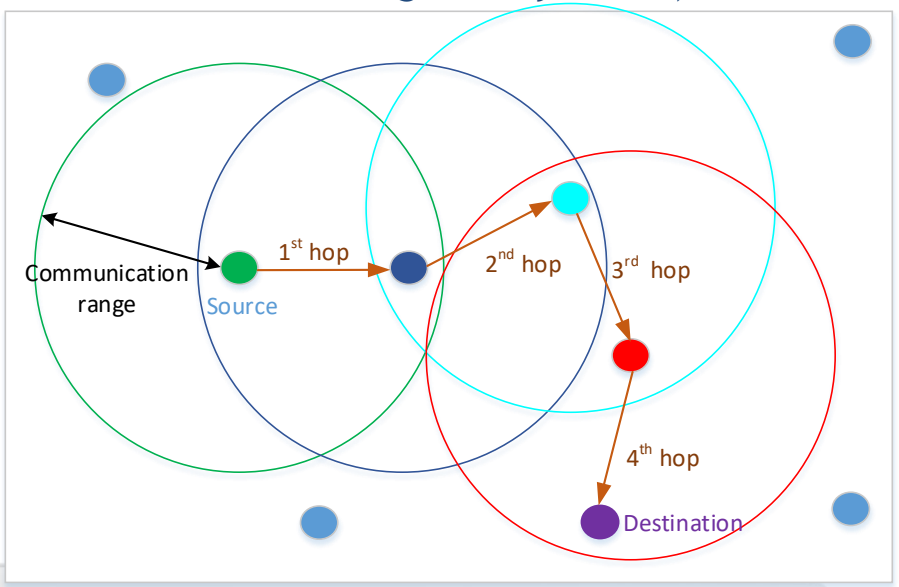
- 1. Source node (sensor) sends a message
- 2. All nodes in the communication range receive the message and retransmits it further to all nodes in their communication range
- 3. This is done as long as the message reach destination or maximum number of hops (TTL) is met

Advantages

- No need to keep any tables, topology or route info

Drawbacks

- Very high overhead (nodes may transmit the same message many times)
- High energy consumption



How to improve flooding?

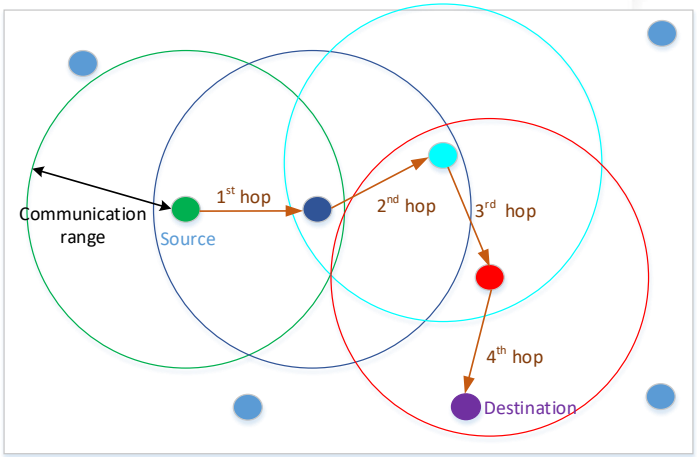
Motivation

- Minimize number of transmitted messages while keep the same probability that message is received at the destination
- Reduce energy consumption

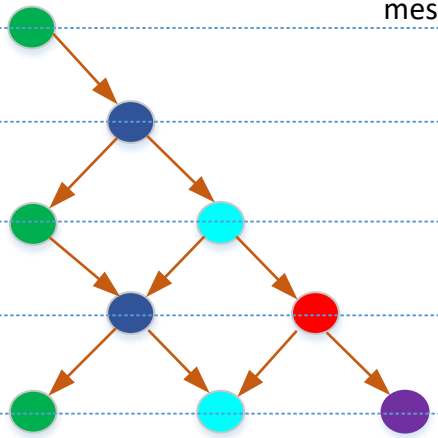
Methods to improve flooding

- **Do not forward the same message again**
  - If the node receives the same message, it is discarded
  - Efficient especially in static scenario where the neighbors do not change
- **Gossiping**
  - Message is forward with just certain probability of *p (p* set to 1 -> flooding*)*
  - Efficient especially if there are huge number of nodes and does not matter if some nodes do not retransmit the message
  - *p* should be properly optimized
    - Low *p* -> take longer to deliver the message (may not be delivered at all)
    - High *p* -> huge number of overhead similarly as in case of flooding
- **Heuristic improvements of gossiping**
  - Hops 1..k => p = 1
  - Hops >k => p < 1
  - Optimize k and p

Illustrative example

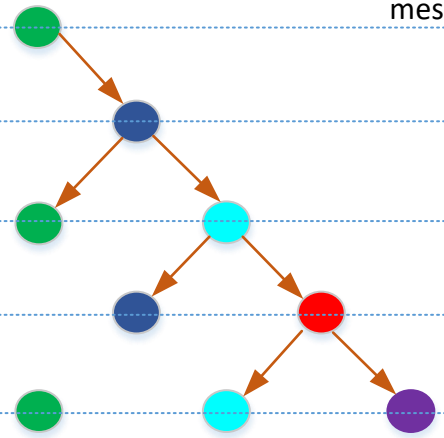


Basic flooding



	No. of transmitting messages (nodes)
1st hop	1
2nd hop	1
3rd hop	2
4th hop	2

Improved flooding (do not forward the same message again



	No. of transmitting messages (nodes)
1st hop	1
2nd hop	1
3rd hop	1
4th hop	1

Objectives

Implement flooding and its improvement(s) in Matlab

Show 2 figures

- Number of generated messages and number of hops depending on communication range (See slide 11 for more details)

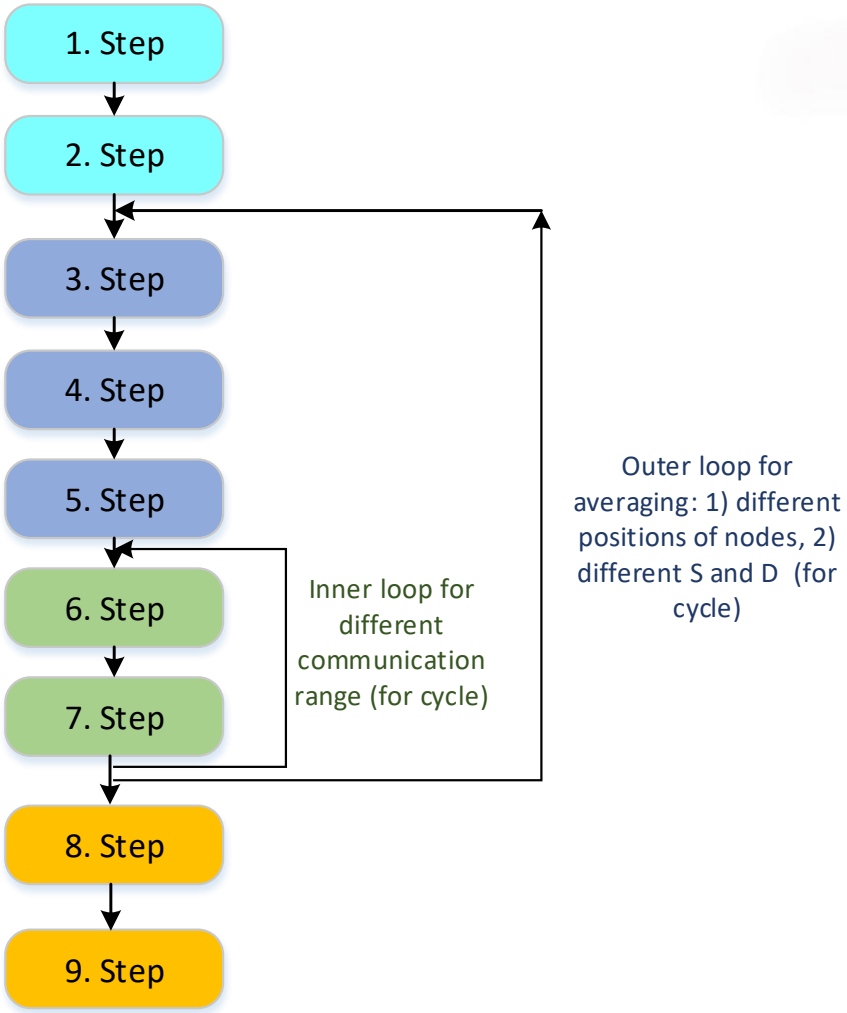
Assumptions

- Simulation area is 500x500m
- Number of nodes (N) is 500
- Communication range between 10 and 100m (one step is 5m -> 19 values)
- Average results over at least 1000 simulation drops
  - Simulation drop
    - Random distribution of nodes in the area
    - Random selection of source and destination nodes (always just one source and one destination node in each drop)

Template

- It is possible to use a template (available at moodle)
- **You can write your own code without the use of template!!**

How to proceed?



How to proceed?

- 1. Define basic parameters
  - **Matlab example:**

```
Simulation_area = [500 500];  % Simulation area in m
No_nodes = 500; % Definition of number of nodes
Sim_drops = 1000;  % Definition of number of drops (for outer loop)
Max_comm_d = 10:5:100; % Definition of max. comm. distance in m (for inner loop)
```

  - Note: Each drop represents different random positions of nodes, S and D
- 2. Define variables to save results
  - **Matlab example:**

```
Total_messages=zeros(19,3);
Total_hops=zeros(19,3);
```

- 3. Randomly generate positions of each node
  - Predefine variable to insert X and Y coordinates of each node (e.g., Nodes_position)
  - Use **"for"** cycle and function **rand**
  - **Matlab example:**

```
Nodes_position=zeros(No_nodes,2);
for node=1:No_nodes
    Nodes_position(node,1)=rand*Simulation_area(1,1); % x coordinates
    Nodes_position(node,2)=rand*Simulation_area(1,2); % y coordinates
end
```

How to proceed?

- 4. Select randomly one source (S) and one destination (D)
  - Use function **rand** to randomly generate indexes of S and D
  - S and D must be different node
  - **Matlab example:**

```
Index_S_D=zeros(2,1); % parameter to store indexes of S and D
Index_S_D(1,1)=ceil(rand*No_nodes); % Index of S node
D=0;
while D==0
    Index_S_D(2,1)=ceil(rand*No_nodes); % Index of D node
    if Index_S_D(2,1)~=Index_S_D(1,1) % Check if S and D are different
        D=1;
    end
end
```

- 5. Calculation of distance between any two nodes
  - **Matlab example:**

```
Node_distance=zeros(No_nodes,No_nodes);
for i=1:No_nodes
    for j=1:No_nodes
        Node_distance(i,j)=sqrt((abs(Nodes_position(i,1)-Nodes_position(j,1)))^2 + (abs(Nodes_position(i,2)-Nodes_position(j,2)))^2);
    end
end
```

How to proceed?

- 6. Set communication range
  - Communication range varies between 10 and 100m
  - **Matlab example:**

```
Max_comm_distance = Max_comm_d(1,CommRange); % Calculation of current comm range
```

- 7. Flooding principles for specific communication range
  - Determine how many messages have been sent
    - Use variable “Total_messages” for saving the number of messages
  - Determine how many hops (time steps) it takes to reach message from S to D
    - Max number of time steps is set to 100 => If message does not reach D in 100 time steps, terminate the cycle (use while cycle)
    - Use variable “Total_hops” for saving the number of hops
- 8. Average the results over K (1000) simulation drops
  - **Matlab example:**

```
Total_messages=Total_messages/Sim_drops;
Total_hops=Total_hops/Sim_drops;
```

- 9. Plot the results

Expected results and rating

Basic flooding (1 point)

Improved flooding (Do not forward the same message again) (1 point)

Gossiping (up to 2 BONUS points)

- Comparison with basic flooding and improved flooding (1 BONUS point)
- Analyze impact of *p* on performance (1 BONUS point)
  - E.g., how *p* should be set with respect to density of nodes in the simulation area

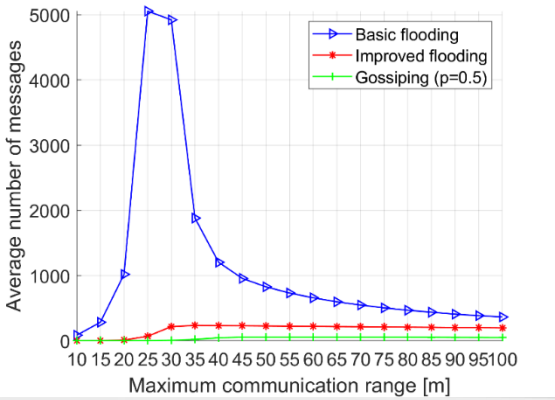
Heuristic optimization of Gossiping (up to 2 BONUS points)

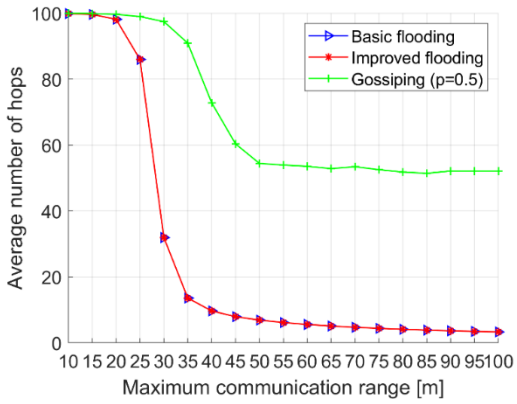
Optional (each student can select what can be done)

Maximum is 4 points!

Evaluation

- During courses
- 4th week of the semester (17.3.)
  - Short discussion on results and code understanding
  - Points given only if student is understanding the code and basic principle of routing mechanisms!!





CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL ENGINEERING

DEPARTMENT OF TELECOMMUNICATION ENGINEERING

Questions?

machp2@fel.cvut.cz