Beacon Vector Routing: Scalable Point-Point Routing in Wireless Sensornets

NSDI 2005
Speaker: Cheng-Han Wu
Outline

• Introduction
  ■ Geographic Routing
    - Physical Coordinates
    - Virtual Coordinates
• The BVR (Beacon Vector Routing) Algorithm
• Simulation Results
• Prototype Evaluation
• Conclusions
Introduction

• Geographic routing
The Disadvantage of Physical Coordinates

- Sensors need to be equipped with GPS.
- It doesn’t work indoors.
- It is inefficient in low node density.
Why is Physical Coordinate Inefficient in Low Node Density?
Why need virtual coordinate?

• It is cheaper than the physical coordinate.

• It can be used indoors.

• It is more inefficient than physical coordinate in low node density
The Virtual Coordinates

- GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks. (INFOCOM)
Virtual Coordinate is More Efficient than Physical Coordinate in Low Node Density
BVR: Beacon Vector Routing

• It requires very little state, overhead or pre-configured information (such as geographic).

• The BVR is implemented on the mica2dot motes.

• The mica2dot motes have several resource constraints—just 4KB of RAM, typical packet payloads of 29 bytes etc.
The BVR (Beacon Vector Routing) Algorithm
• Construct the virtual coordinate

• Greedy forwarding over node coordinates

• If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.

• A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.

![Diagram]

- : beacon
- : source
- : destination

(A,B,C)
• Construct the virtual coordinate

• Greedy forwarding over node coordinates

• If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.

• A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.

![Diagram showing the process]

- : beacon
- : source
- : destination
How to Construct the Virtual Coordinate?

Coordinate=(A,B,C)
Coordinate=(1,3,2)

: beacon
• Construct the virtual coordinate

• **Greedy forwarding over node coordinates**

• If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.

• A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.

\[ \text{A} \text{: beacon} \]

\[ \text{B} \text{: source} \]

\[ \text{C} \text{: destination} \]
Greedy Forwarding over Node Coordinates

- If there are $r$ beacons, a node forwards the packet by only considering $k$ closest beacons to destination. ($k<r$)
Greedy Forwarding over Node Coordinates

\[ \delta^+_k (p, d) = \sum_{i \in C_k(d)} \max(p_i - d_i, 0) \quad \text{and} \quad \delta^-_k (p, d) = \sum_{i \in C_k(d)} \max(d_i - p_i, 0) \]

\( P \): current routing node \quad \text{\( d \)}: destination

\[ \delta_k = A \delta^+_k + \delta^-_k \], \( A \) is constant (assume \( A=10 \))

\[ \delta^+_k = \max(4-3,0) + \max(6-1,0) = 1 + 5 = 6 \]

\[ \delta^-_k = \max(3-4,0) + \max(1-6,0) = 0 \]

\[ \delta_k = 10 \times 6 + 0 = 60 \]
Greedy Forwarding over Node Coordinates

\[ \delta^+_k(p, d) = \sum_{i \in C_k(d)} \max(p_i - d_i, 0) \quad \text{and} \quad \delta^-_k(p, d) = \sum_{i \in C_k(d)} \max(d_i - p_i, 0) \]

- \( P \): current routing node
- \( d \): destination

\[ \delta_k = A\delta^+_k + \delta^-_k \]

A is constant (assume \( A=10 \)).
• Construct the virtual coordinate

• Greedy forwarding over node coordinates

• If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.

• A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.
Simulation Results

• Assumptions:
  ■ Each sensor has a fixed circular radio range

  ■ A node can communicate with all and only those nodes that fall within its range.

  ■ The simulator ignores the capacity of, and congestion in the network.

  ■ The simulator ignores packet losses.
Figure 1: Success rate of routes without flooding in a 3200 node network, for different numbers of total beacons, $r$, and routing beacons, $k$. 
The Impact of Node Density

![Graph showing the impact of node density on success rate without flooding.](image)
The Impact of Node Density

![Graph showing the impact of node density on success rate without flooding. The y-axis represents the success rate, and the x-axis represents the number of beacons. There are four lines representing different conditions: True positions, high density (o.d. 2-hop), High Density (15.7), Low Density (9.8), High Density + on-demand 2-hop (17.0), and Low Density + on-demand 2-hop (12.7).]
How many beacons do we need?

Figure 4: Number of beacons required to achieve less than 5\% of scoped floods, with $k = 10$ routing beacons.
Prototype Evaluation

Two testbeds:

- (Office-Net) consists of 42 mica2dot motes in an indoor office environment of approximately 20X50m.
- (Univ-Net) is a testbed of about 74 mica2dot motes deployed across multiple student offices on a single floor of UC Berkeley’s Computer Science building.
Univ-Net

![Graph showing the percentage of routes over time. The graph includes lines for Success (Greedy + Scoped Flood), Scoped Flood, Contention, Failure, and Route Request Rate. The x-axis represents time in minutes, ranging from 0 to 120, and the y-axis represents the percentage of routes from 0 to 140. The graph shows a significant increase in route requests per second around the 110-minute mark.]
Conclusions

• Beacon Vector Routing is a new approach to achieving scalable point-to-point routing in wireless sensor networks.

• The advantage of BVR are its simplicity, making it easy to implement on resource constrained nodes like motes.