Node Filtering and Face Routing for Sensor Network

Umang Amatya
Department of Computer Science

Main Contributions
- Efficient Algorithms for identifying Redundant Sensor Nodes
- New Technique for Filtering Redundant Nodes in Sensor Network
- Reliable Algorithm for Message Routing - Forwarding
- User Friendly Prototype Implementation in Java
- Results of Experimental Investigation

What are Sensor Nodes?
- Small scale electronic devices with
  - Limited memory and processing power
  - Wireless Communication Interface (range ≈ 70 meters)
  - Low-power Battery
  - Sensing component (temperature, sound, …)
- Usually assumed to be stationary
- Allowed to be active/inactive

Typical Sensor Nodes:
- Berkeley MICA Mote
  - Cost ≈ $10
  - Size ≈ 40cm³
  - Weight ≈ 70g
  - Battery ≈ 15k
  - Memory ≈ 4KB Ram
  - OS – TinyOS
  - Radio Range ≈ 30m
- Smart Dust
  - Cost ≈ $1
  - Size ≈ 0.01cm³
  - Weight ≈ 0.0201g
  - Battery – Less
  - Memory – Less
  - OS – Smaller
  - Radio Range – Shorter
- WINS NG 2.0
  - Cost – $100
  - Size – 1300cm³
  - Weight – 5400g
  - Battery – 300k
  - Memory – 32MB
  - OS – Linux
  - Radio Range – 100m

Sensor Network Modeled by Gabriel Net

Linking Rule: Given a set of nodes S = p₁, p₂, p₃, …, pᵢ, two nodes pᵢ and pⱼ are connected by an edge if the disk with diameter ending at pᵢ and pⱼ does not contain any node.

Greedy Routing in Gabriel-Net
- How to send packet from the source(S) node to destination(D) node?
- Greedy Rule: Forward packet to the node closest to the destination node
- Thick-Segment Route ( ———- ) Path Traced by Greedy Rule

How to Fix Trapping Problem?
- Follow the Face of the Network to exit from the Trap (shown by directed line-segments)

Identifying Redundant Nodes
- Two nodes close to each other are called equivalent if their transmission range cover the same sub-set of nodes.

Characterizing Floating Chains (enclosed by thick ellipse)

Network after Removing Floating-Chains

First Filtering Rule: Just keep representative node from redundant clusters (black-nodes)

Refined Network after Removing Redundant Members (un-connected!)

First Result (Our Finding): Connectivity between S and D is not compromised by removing redundant nodes

Second Result: Solo-Faced Floating Chains can be Filtered without compromising connectivity between S and D

Characterizing 2-connected components
- External 2-connected components can be removed without compromising Network Connectivity between S and D

Applying All Three Filtering Rules
- Network after filtering External 2-Connected Components. Filtered nodes are shown as tiny background dots.

Java Implementation
- Greedy Routing: Successful most of the time.
- Hybrid Greedy-Face Routing: Guaranteed to be successful if there is a path from source to target
- Node Filtering: Used for extending battery life.

Future Work and Extensions
- We are picking a redundant member from equivalent nodes arbitrarily
- Develop Rules for picking appropriate node from equivalent cluster
- Extensive simulated experimental investigation on randomly deployed sensor nodes
- Apply Proposed Filtering Method on Real-World Sensor Network

Randomly Generated 500 Sensor-Nodes inside a Rectangular Region

Randomly Generated 600 clustered Sensor-Nodes inside a Rectangular Region

Occasional Trouble with Greedy Routing!
- While sending packet from red node (S) to blue node (D), it get trapped in the green node (T)

What are Sensor Nodes?
-Sensor Network Modeled by Gabriel Net
-Greedy Routing in Gabriel-Net
-How to Fix Trapping Problem?
-Identifying Redundant Nodes
-Characterizing Floating Chains (enclosed by thick ellipse)
-Network after Removing Floating-Chains
-Java Implementation
-Future Work and Extensions

Sensor Network Modeled by Gabriel Net

Greedy Routing in Gabriel-Net

How to Fix Trapping Problem?

Identifying Redundant Nodes

Characterizing Floating Chains (enclosed by thick ellipse)

Network after Removing Floating-Chains

Java Implementation

Future Work and Extensions