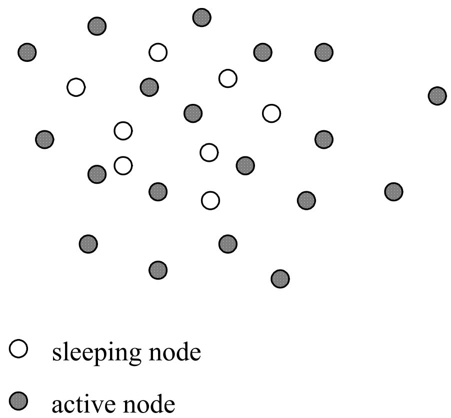
Responsibility Region Allocation for Dense Sensor Networks

Introduction

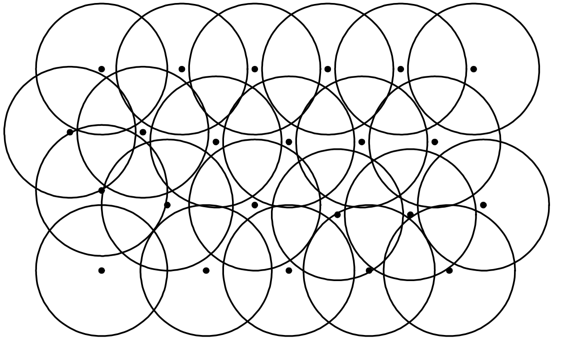
- Energy is a scarce resource for a sensor device
- A randomly deployed dense sensor network → many redundant nodes
- To conserve energy, previous research focuses on:
  - Turning off redundant nodes to prolong network lifetime
  - Aggregate same information to reduce the number of transmission

Previous Techniques

- Turning off redundant nodes
  - Unfortunately the overlapped area may still be considerable
  - Turning off some nodes may result in isolated sub-network



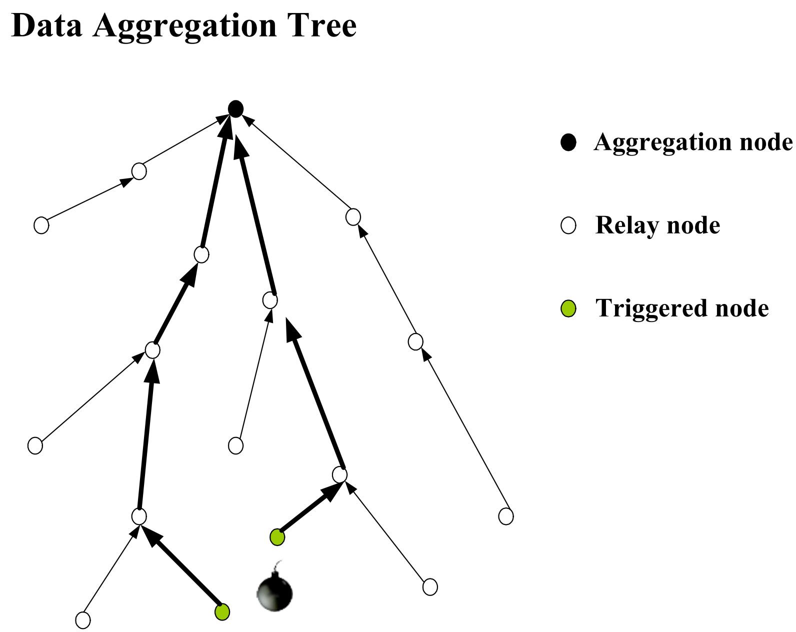
(a)



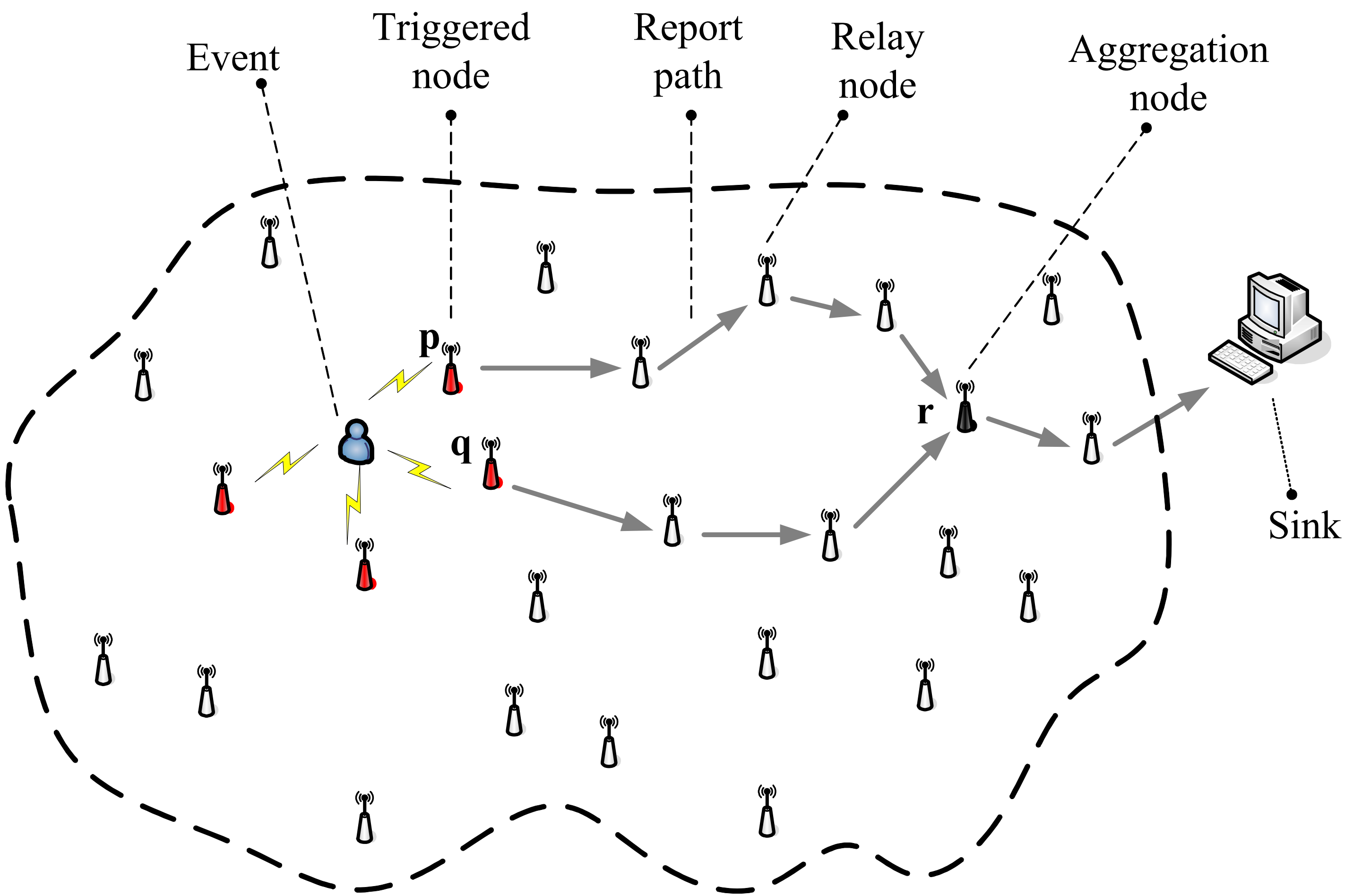
(b)

Previous Techniques

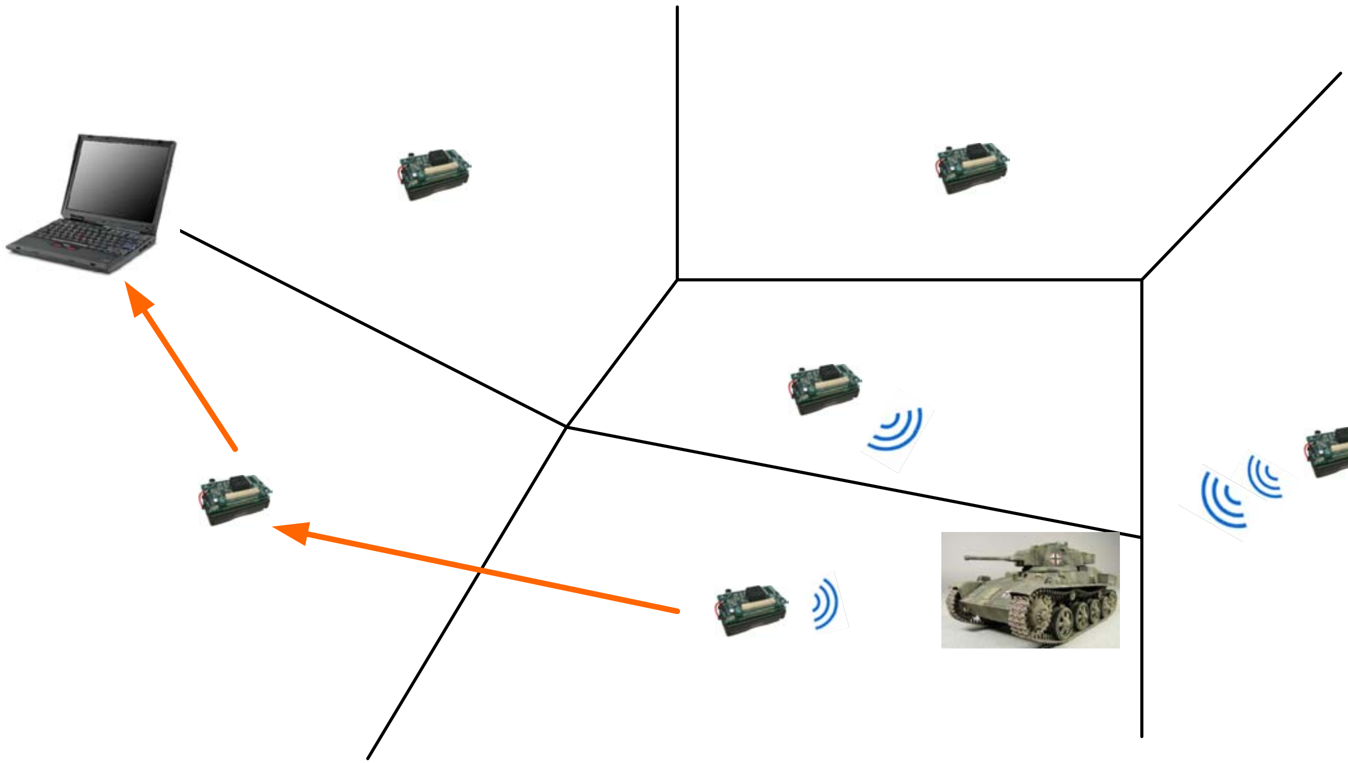
- Data aggregation
  - To reduce the number of packet transmission
  - However, the aggregation node may be far away from event-triggered nodes
  - Cost of tree, or graph, construction
  - Aggregation overhead increases packets delay



Previous Techniques



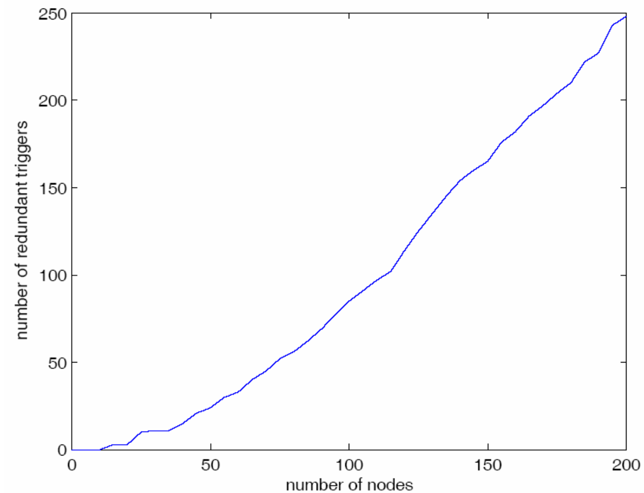
Motivating Example

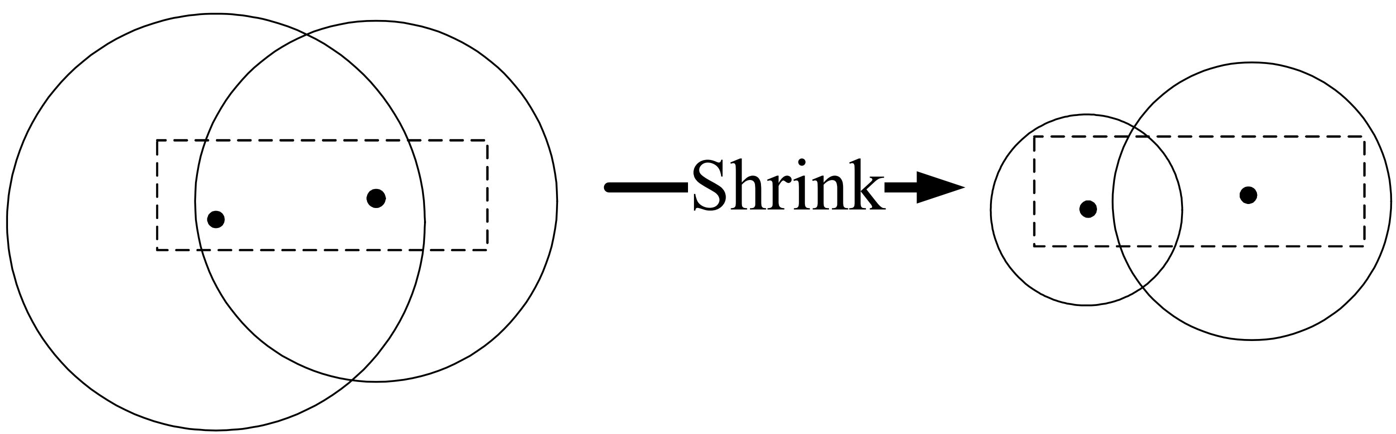


- Each sensor is in charge of one specific region

Motivation

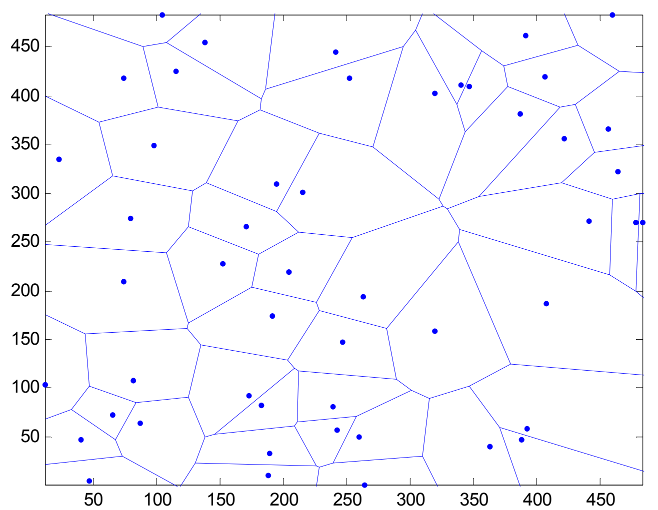
- Each sensor has an unique guarding zone
  - To minimize redundant triggers
  - To decrease packet delay
  - To conserve energy
    - Less transmission less energy consumption
  - To alleviate packet collision



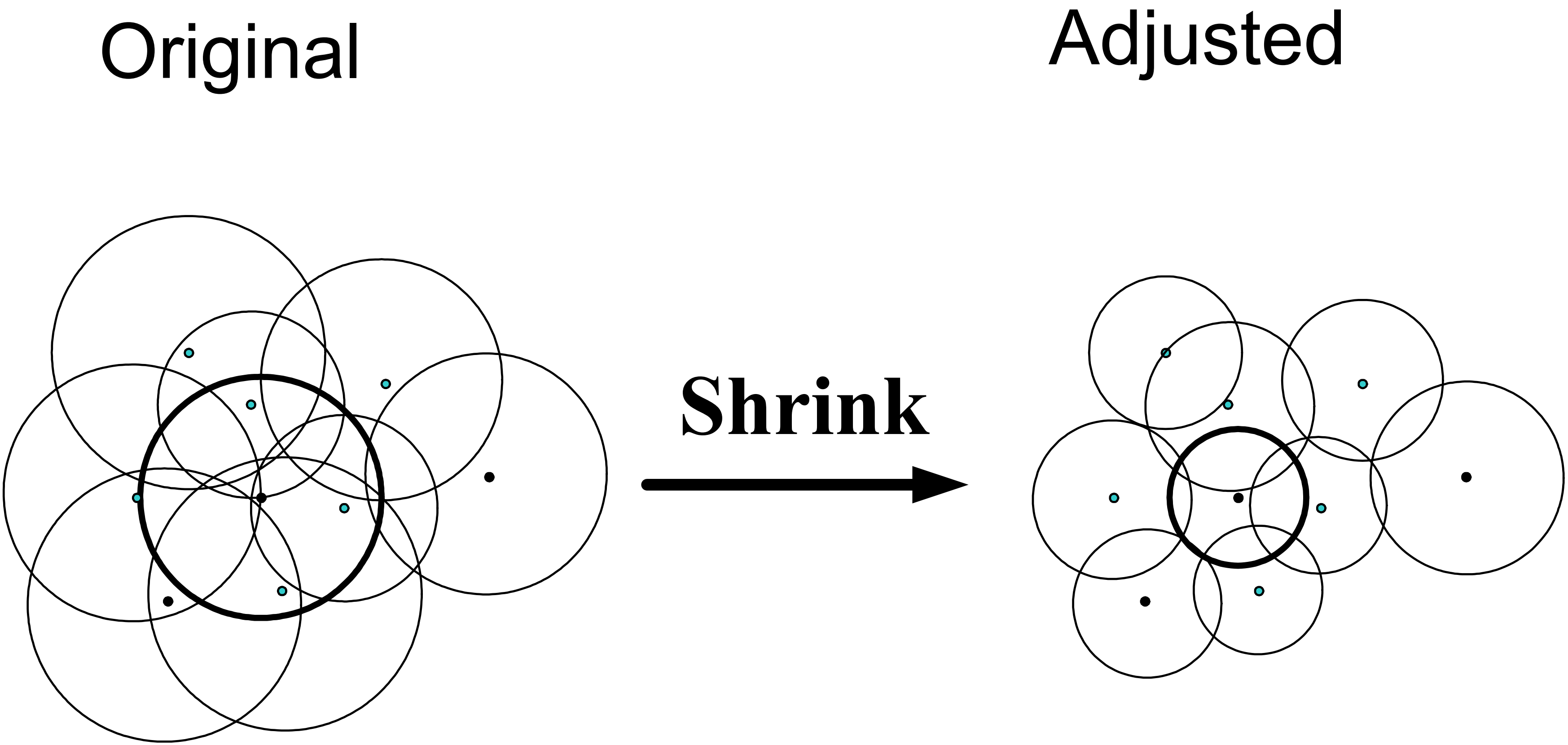


How to allocate responsibility regions?

- The responsibility region can be any irregular shape
- Voronoi Diagram
  - Need to know the event location in advance
- Assume that we only know event distance
  - Associate each sensor with a responsibility radius
  - sensor reports *e* if *e* is within responsibility radius



A shrinking example



Assumption

- Sensing radii of sensors are identical
- Sensor's location is known
- Event distance can be estimated by evaluating event strength

Requirement & Criterion

- Sensors can self-organize their specific responsibility region based on their local states
- No event loss
- Size of responsibility region should be related to remaining energy of sensors.

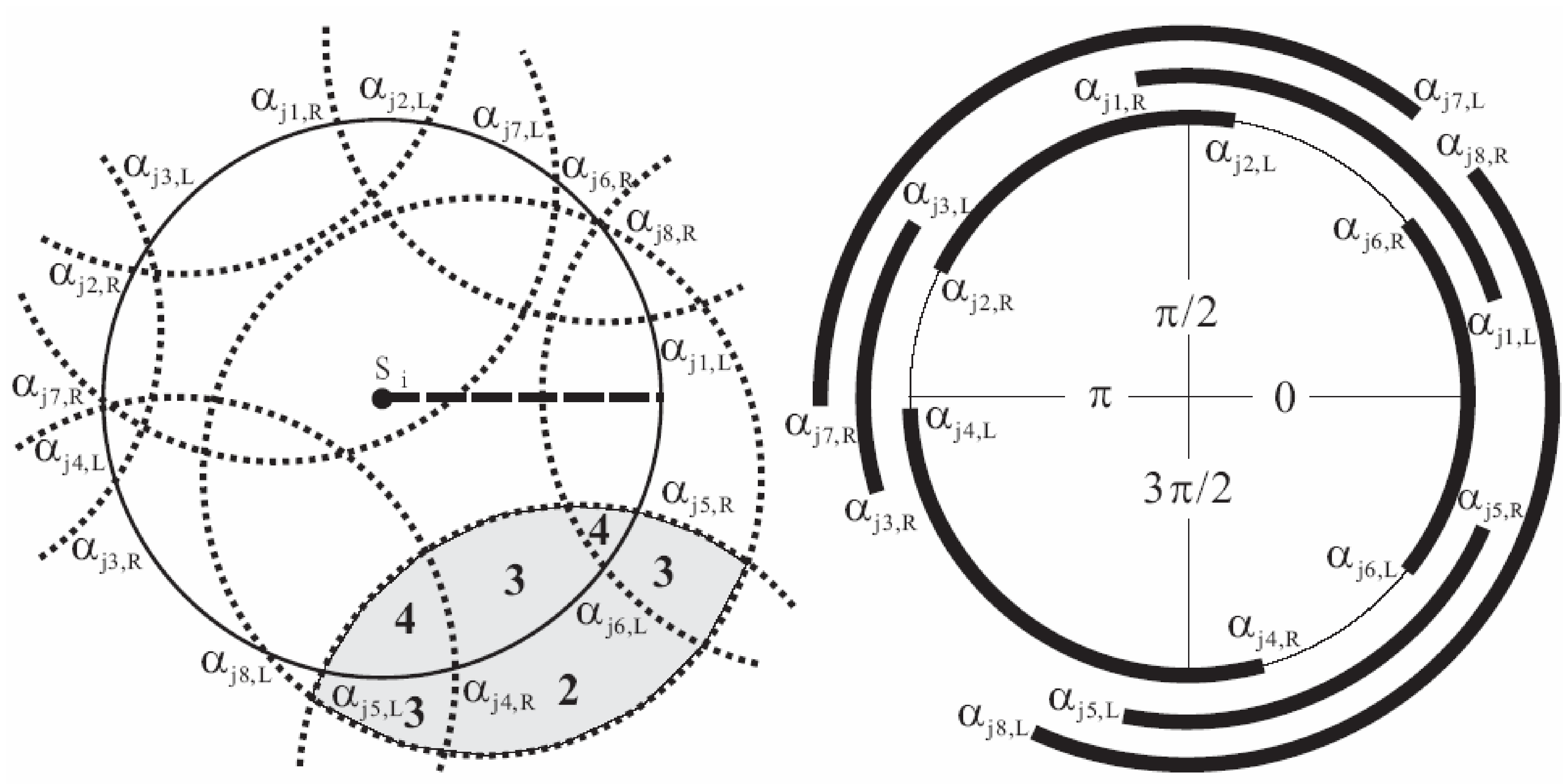
Problem Definition

- Definition (Responsibility radius), Rg:
  - a virtual sensing radius of a sensor, i.e., events beyond Rg should be ignored by that sensor.
- We are given a sensor network S = {s1, …, sn}, deployed over a two-dimensional area *A* and associated with an initial Rg vector (Rg^1, Rg^2,…,Rg^n). We want to find a new Rg vector, say R’g, such that the aforementioned requirement is met.

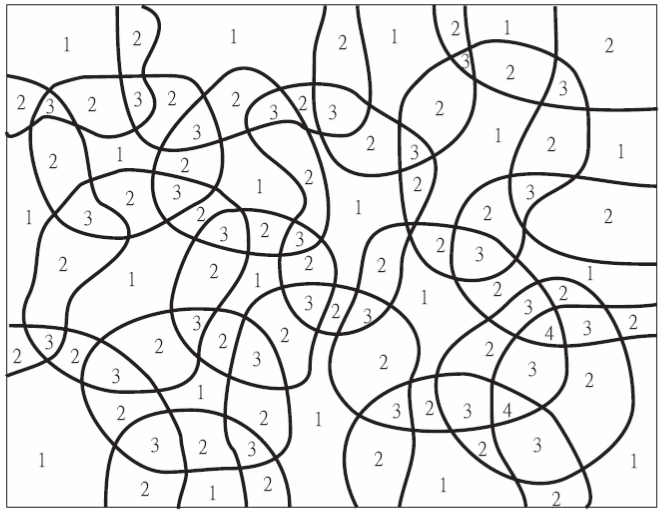
Token-based basic algorithm

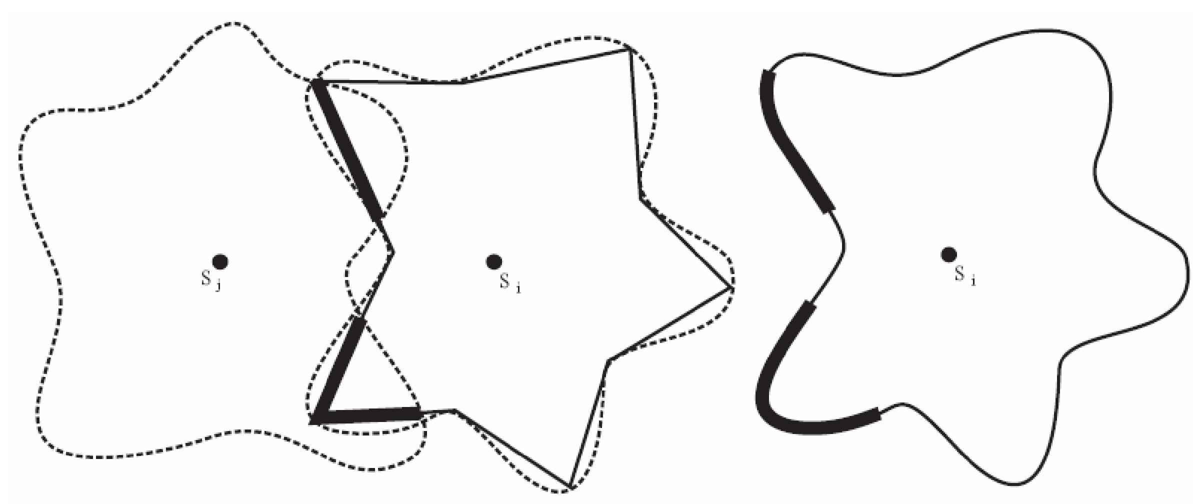
- Rg^i is divided into *x* segments.
- To avoid collision, sensors randomly start and broadcast a message to indicate getting the token
- Let $\delta_i = Rg^i / x$. Sensor who gets token shrink δ_i once at a time if the new Rg do not make any coverage hole.
- After shrinking, broadcast a message to indicate releasing the token.
- The basic algorithm terminates if no sensor can shrink anymore.

1-perimeter-covered (circle)



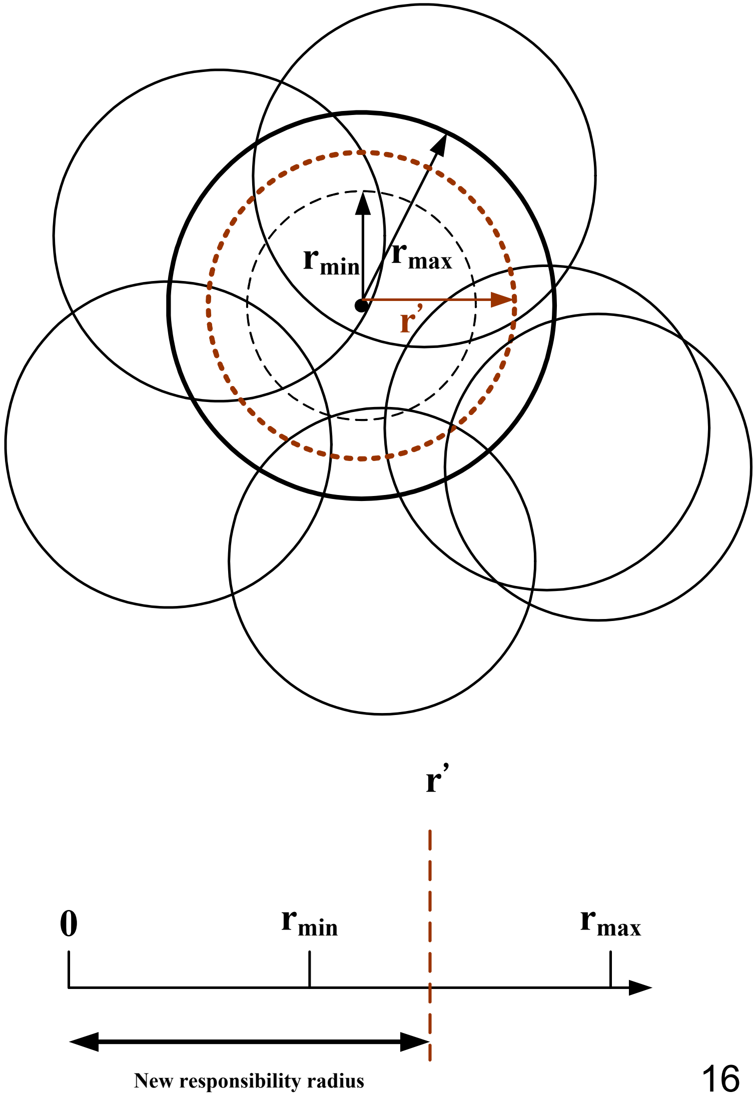
1-perimeter-covered (irregular shapes)





Token-based greedy algorithm

- Sensors who get token execute the following steps:
  - Eliminate sensors which do not contribute coverage
  - Calculate r_{min} , choose appropriate r' value.
  - R'g = r'
  - Broadcast a message to release token



Virtual force-based algorithm

- Faster than token-based algorithms
  - All sensors can run at the same time
- A sensor responsibility radius is like a rubber band
  - Fin: contraction force = Rg square
  - Fout: neighbors' pulling force, related to neighbors' distance and neighbors' Rg
- Let Fin = Fout → get new Rg

Discussion

- With sensor location and event location
  - Voronoi Diagram
  - Difficult to retrieve event location info.
- Without sensor location
  - Modified PEAS (Probing Environment & Active Sleeping)
  - Some events may lose
- With Sensor location and event strength
  - reasonable scenario
  - With event strength, sensors can estimate the distance of the event