Coverage-oriented Network Scheduling and Location-directed Data Collection
Towards Energy-efficient Wireless Sensor Networks

(Draft Version)

ZHOU, Yangfan

Department of Computer Science & Engineering
The Chinese University of Hong Kong
September 25, 2009

Outlines

- Introduction to wireless sensor networks
- Part 1: Network partitioning for reducing coverage redundancy
- Part 2: Network partitioning for maximizing coverage
- Part 3: Optimizing data forwarding path
- Part 4: Contour mapping for in-network data preprocessing
- Conclusions

Introductions: Sensor Nodes

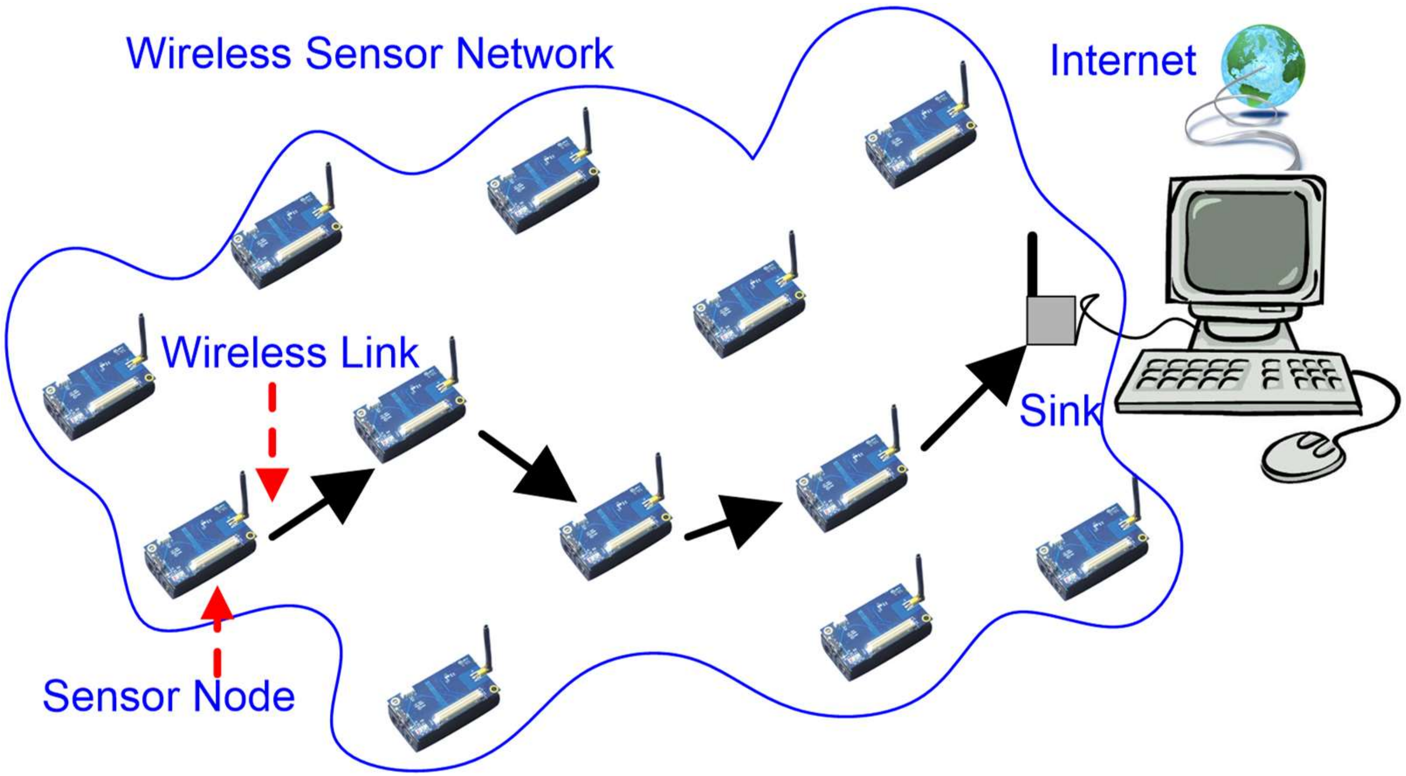
Wireless-Integrated Embedded Sensor Device

- Sensors
  - e.g., infrared, acoustic, vibration, barometric, humidity, thermoelectrical, photosensitive
- Micro-computer system
  - CPU: several to tens of MHz
  - Memory: several KB
  - Storage: flash memory: e.g., 128KB
- Wireless interface
  - IEEE 802.15.4: CSMA
  - Low-power: tens of meters
- Battery: may not be rechargeable or replaceable

Achieve energy-efficiency with limited computational (speed and memory capacity) resource

Introductions: Wireless Sensor Networks

Wireless Sensor Networks (WSNs)



Introductions: WSN Applications

Applications of Wireless Sensor Networks (WSNs)

- Environmental monitoring
- Environmental measurements
- Habitat monitoring
- Drink water quality monitoring
- Industrial process monitoring
- Structure health monitoring
- Intruder detection
- Vehicle classification and tracking
- ......

Introductions: Topics in This Thesis

Applications of Wireless Sensor Networks (WSNs)

Event Detection → **Alarm when an event of interest is detected**

- A fire (a high temperature)
- An intruder (certain kind of sound)

Part 1: Reduce redundancy
Part 2: Maximize coverage

Data Collection → **Stream the sensor data to the sink**

- Distribution of temperature
- Intensity of the sunlight

Part 3: Optimize the forwarding path
Part 4: Preprocess data in network

- Reduce the forwarding packet number
  - Location-centric approaches

Network Partitioning (Reducing Redundancy)

Coverage-oriented network partitioning

Part 1: Redundancy Reduction

Part1: Problem Motivation and Formulation

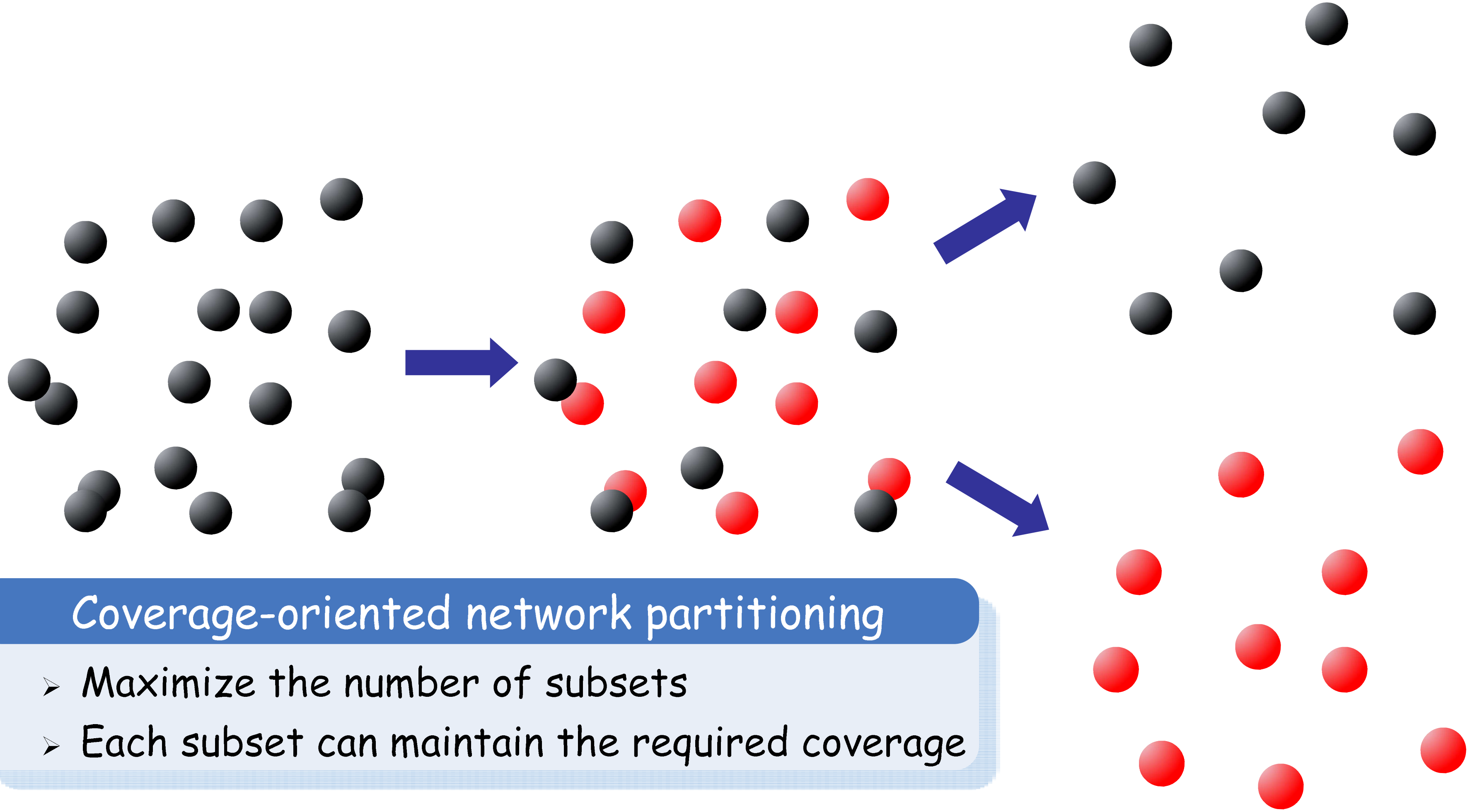
Motivations

- Fault Tolerance
  - WSNs contain a large number of sensor nodes
  - A small number of these nodes are enough to provide entire field coverage
- Energy-efficiency
  - Exploit the redundancy
  - Put those redundant nodes to sleep mode

Network partitioning problem

- Divide the sensors into disjoint subsets
  - Each subset can provide entire network coverage
  - Schedule the subsets so that they can work successively

Part 1: Problem Illustration



Coverage-oriented network partitioning

- Maximize the number of subsets
- Each subset can maintain the required coverage

Part 1: Existing Algorithms

Algorithms to be compared

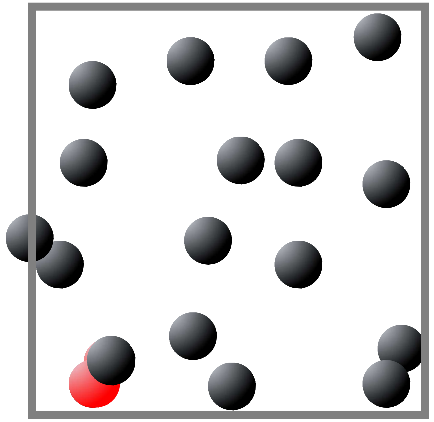
- Algorithm based on mixed integer programming proposed by M. Cardei *et al* (MIPA)
  - Model Sensor Grouping Problem as a Disjoint Set Covers (DSC) problem which is NPC
  - Transform the DSC problem to a maximum-flow problem with a mixed integer programming (MIP) formulation
  - Solve the MIP with an MIP solver
- Greedy Algorithm (GA)
  - Key idea: select an ungrouped node to the current subset so that the area covered through adding this node is maximized comparing with adding any other ungrouped nodes

MIPA can find more subsets, but it is much slower

Part 1: Intuitions to Problem Solution

Intuitions

- Given a coverage redundant network
  - For all ungrouped nodes, select them all into the current subset
  - Remove node one by one, until breaking the coverage criteria



These two are the closest → Their covering areas are similar to each other → redundancy!!!

Remove the one farther to the other nodes → The rest of the nodes will be closer to each other → More nodes can be removed later, potentially

Part 1: Introducing the Normalized Minimum Distance

Analysis

- Find the closest node pair and remove a node in the pair
  - Effect: Increase $\min_{\forall i,j \& i \neq j}(||x_i - x_j||)$ of the rest
    x_i: The location of each point
- Remove the one farther to the other nodes, i.e., the one with larger average distance to the other node
  - Effect: Reduce μ, i.e., the average distance between nodes

---

- Hence, we adopt the following heuristic
  - Introduce a node fan-out index

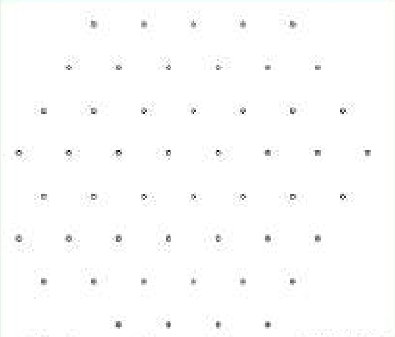
$$\iota = \frac{\min_{\forall i,j \& i \neq j}(||x_i - x_j||)}{\mu}$$

  - Remove the node so that the rest of the nodes achieve maximum ι

Part 1: Property of the Index

Property of the index

Many many points



Node number = 40, ι = 0.324970

- If the points are mobile, maximizing ι results in an equilateral triangle structure
  - *i.e*, the Voronoi diagram is honeycomb-like
  - The coverage efficiency is the best
- $\therefore$ In coverage-related problem, maximizing ι is a promising approach to exploit redundancy

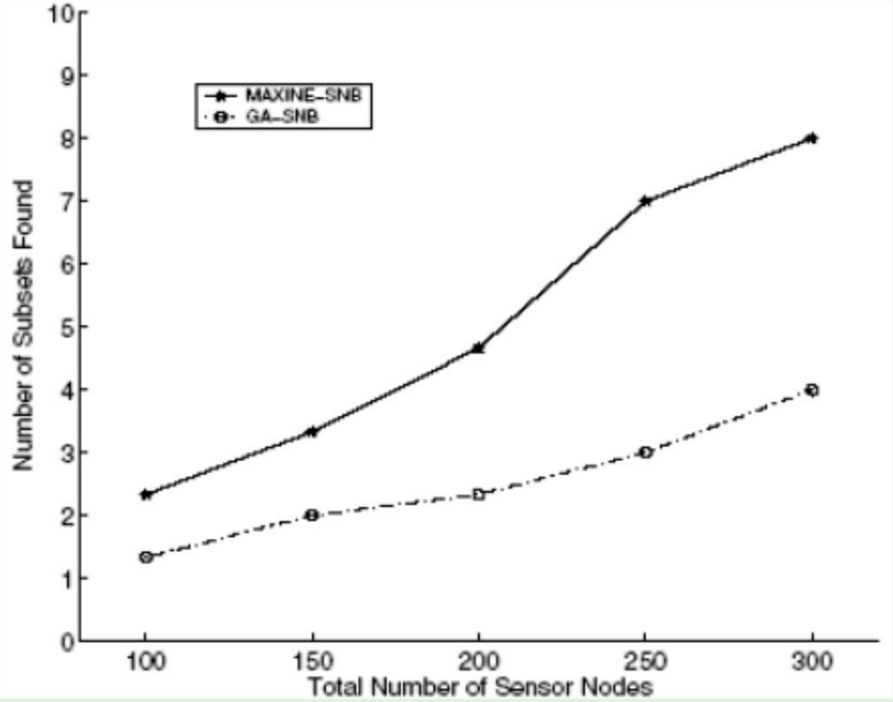
Part 1: Our MAXINE Algorithm

Maxine: MAXimizing-ι Node-redundancy Exploiting

- Basic idea
  - The ι of the resulting subsets should be large
  - So, we try to greedily maximize it...
- Process the following procedure to find a subset until eventually, selecting all ungrouped nodes to the current subset cannot guarantee entire field coverage
  - Tentatively selects all ungrouped nodes into the current subset
  - One by one, remove nodes from the subset until the deletion of any of the nodes in the subset will result in uncovered field
    - The deletion of the selected node does not result in any uncovered area
    - The deletion of the selected node results in maximum ι value of the current subset, comparing to the deletion of all other nodes in the subset
  - Finish processing the current subset, begin to process next subset.

Part 1: Simulation Studies

The number of subsets found by Maxine and GA



Maxine performs much better than GA in terms of the number of subsets found

Part 1: Simulation Studies

The number of subsets found

	$x = 16$		$x = 36$	
n	MIPA	MAXINE	MIPA	MAXINE
75	2.00	2.00	1.67	1.33
100	4.00	4.00	4.00	3.33
125	5.33	5.33	4.67	4.33
150	7.67	7.33	6.67	5.33
175	9.00	8.33	8.67	6.67

	$x = 64$		$x = 100$	
n	MIPA	MAXINE	MIPA	MAXINE
75	1.33	1.33	1.33	1.33
100	2.67	2.67	3.00	2.67
125	4.00	3.33	3.67	3.00
150	6.00	5.00	5.67	4.67
175	7.00	6.00	7.33	5.00

	$x = 144$		$x = 196$	
n	MIPA	MAXINE	MIPA	MAXINE
75	0.67	0.67	1.00	1.00
100	3.00	2.67	3.00	2.33
125	3.67	2.67	4.00	3.00
150	6.00	4.67	6.00	4.67
175	6.00	4.33	6.33	4.67

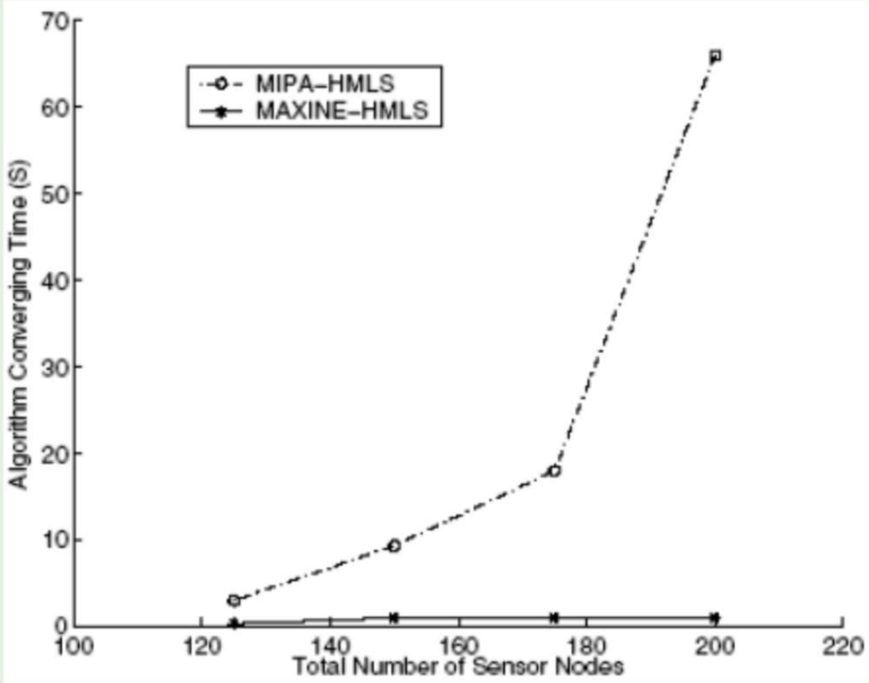
Total in-network node number

Total sampling points
We use quasi-random sequences to select a set of points in the network area. The network is considered covered if all the points are covered

- The number of subsets found by MIPA and Maxine is comparable

Part 1: Simulation Studies

The impact of sensor node numbers on converging-time



Maxine's converging time is much shorter than MIPA

Part 1: Simulation Studies

Simulation Conclusions

- Maxine and GA converge more quickly than MIPA does
- The performance of Maxine and MIPA is comparable in terms of # of subsets found
- The performance of Maxine is much better than GA in terms of # of subsets found

Part 1 Contributions

- Solve a coverage-oriented network partitioning problem which is critical for energy saving. Our approach is fast and the results are satisfactory.
- Propose a fan-out index and show that maximizing the index can increases coverage efficiency

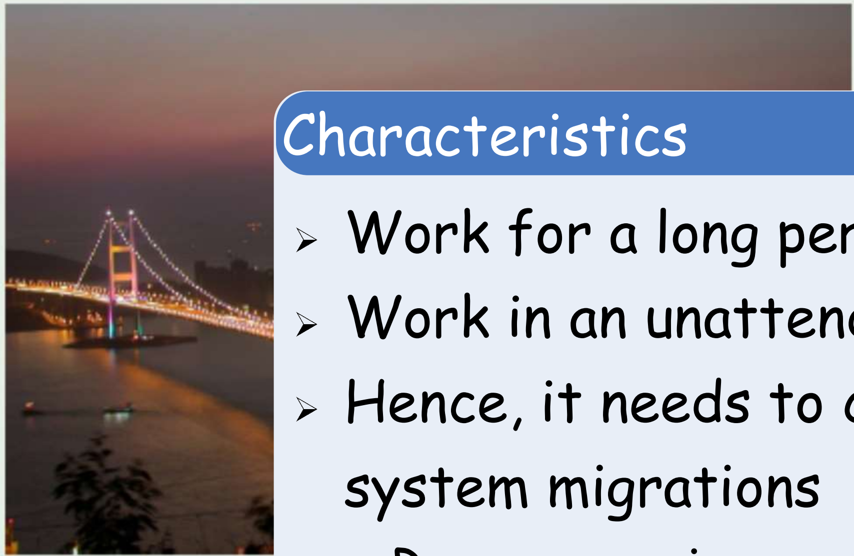
Network Partitioning (Maximizing Coverage)

Coverage-oriented network partitioning

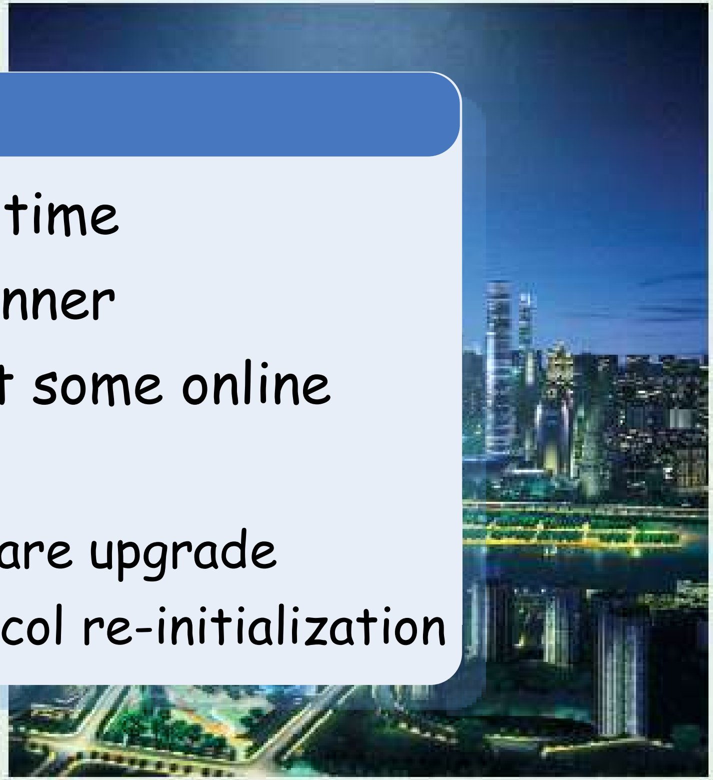
Part 2: Coverage Maximization

Part 2: Background

Tsing-Ma bridge



Guangzhou TV tower



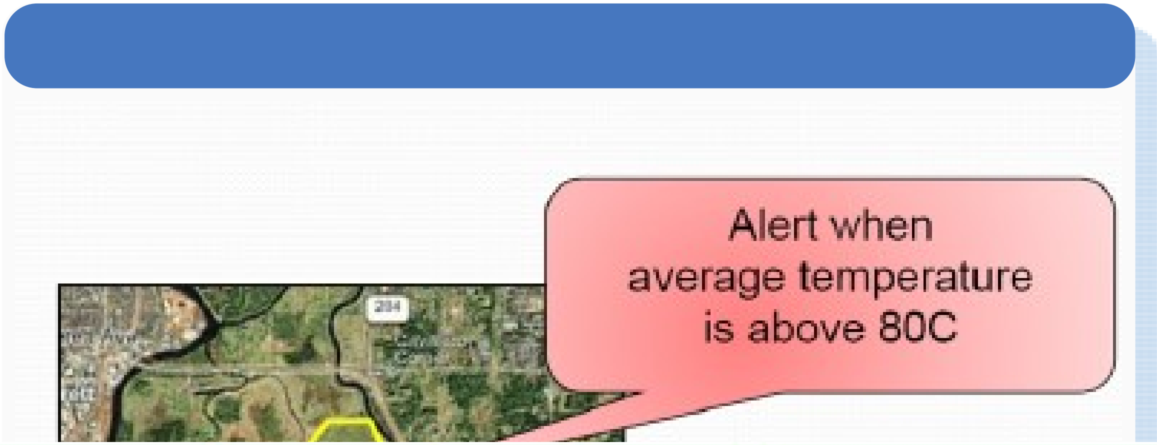
Characteristics

- Work for a long period of time
- Work in an unattended manner
- Hence, it needs to conduct some online system migrations
  - Reprogramming, e.g., software upgrade
  - Re-initialization, e.g., protocol re-initialization

Part 2: Motivations

Characteristics

- Much work has been conducted to provide online system migration support



The sensor network configuration problem

How to divide the network into N subsets so that the event detection capability of the network during a system migration is maximized.

Now the problem becomes:

How to divide the network into N subsets?

Part 2: Problem Formulation

At least one sensor in the working division can detect an event located at (x, y)

$$p_D(x,y) = 1 - \prod_{i=1}^{n}(1 - c_i p_i(x,y))$$

- QoS is modeled as the event detection probability by the working sensors when a subset is performing a migration task.
- n: sensor numbers
- $\{s_i\}_{i=1}^{n}$: sensor nodes
- D: the set of the working sensors
- (x,y): event location
- $P_i(x,y)$: the probability that sensor s_i detects the event
- c_i: whether sensor s_i is in D: 1 if *yes*, 0 otherwise

We name it a working division. Subset j is doing migration, the rest of the nodes are called working division j

The probability that a sensor detects an event

The probability that a sensor CANNOT detect an event

The probability that all sensors cannot detect an event

Part 2: Problem Formulation

$$P_D = \min_{\forall (x,y) \in \phi} p_D(x, y)$$

- A pessimistic measure: the worst case is representative

Part 2: Baseline Algorithms

Random Pick (RP)

- Randomly select n/N nodes for each subset
- RP is a baseline algorithm to see how well (or how badly) other algorithms perform.

Greedy Algorithm (GA)

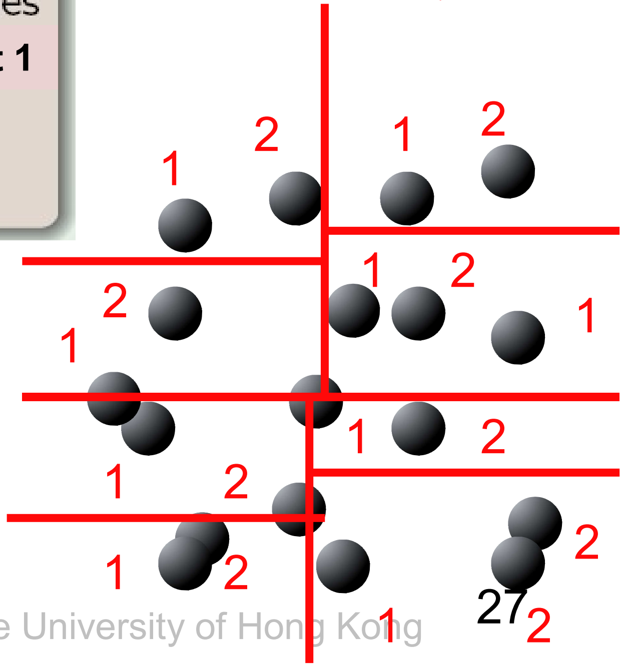
- First perform the RP algorithm
- Perform the following iteratively
  - Find p_{min}, locate t_x and find D_y
    - p_{min}: the minimum event detection probability among the event detection probabilities of any division at any sampling points
  - Find S_z so that the event detection probability of S_z is the minimum among all S_k at t_x
  - Move a node from S_y to S_z so that p_{min} improves the most. We are done if p_{min} cannot be improved

Part 2: Simple Partitioning and Picking (SPP)

Partitioning Procedure

- Consider all nodes are in one region, and perform the following iteratively until there are less than $2N$ nodes in every region
  - For each region, draw a line parallel to the x-axis to partition the region into two so that the number of nodes in each partition is the **same or their difference is at most 1**
  - Then for each region, draw a line parallel to the y-axis to partition the region into two so that the number of nodes in each partition is the **same or their difference is at most 1**
- Nodes in each region are randomly selected into N different subsets

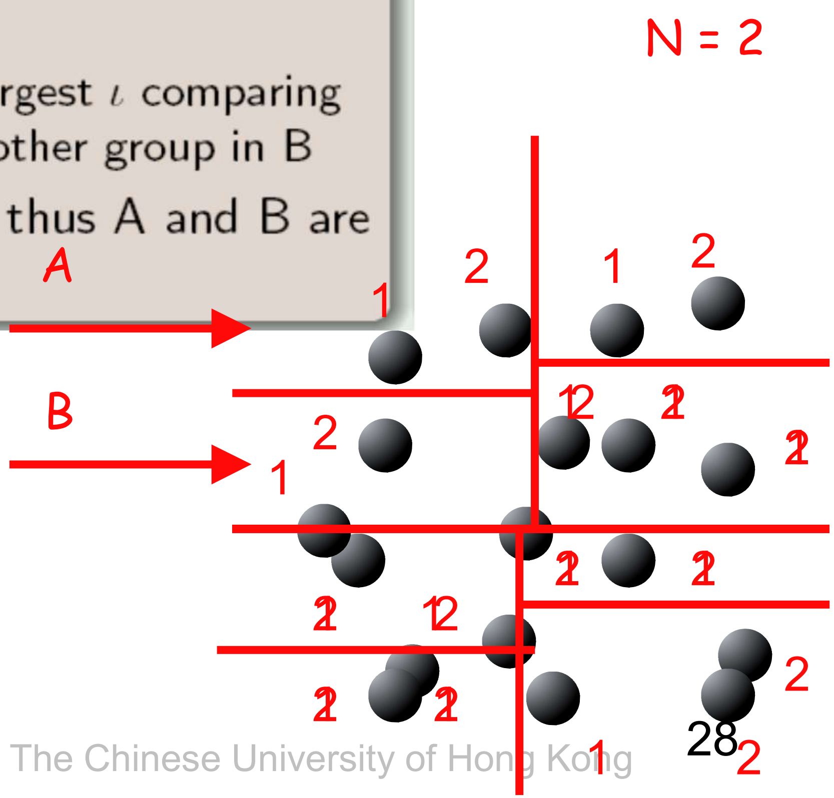
N = 2



Part 2: The SPP Algorithm

Merging Procedure

- Neighboring regions are merged into one till there is only one region as follows.
  - Randomly assign two neighboring regions as A and B
  - Couple each randomly picked group (say A1) in A and a selected group B_i in region B. The selection criteria of B_i is:
    - The couple (A_1 & B_i) can result in largest ι comparing to the couple formed by A1 and any other group in B
  - Each couple is deemed as a subset, and thus A and B are merged into a larger region.



The underlying idea of SPP is to let each subset distributed evenly in the network, so that when one ceases to work, it makes a minimum impact on the event detection capability.

Part 2: Minimum Spanning Tree-based Grouping (MSTBG)

Minimum Spanning Tree-Based Grouping Algorithm (MSTBG)

- First build a tree which is composed only by two nearest nodes among all the in-network nodes and randomly assign these two nodes to different subsets.
- Perform the following iteratively until all nodes are in the tree
  - Select a node nearest to the tree but not in the tree
  - Group the node to its farthest subset. Then add this node to the tree.
    - The distance between a node n and a subset S is defined as the distance between n and the nearest node in the tree which is in S.

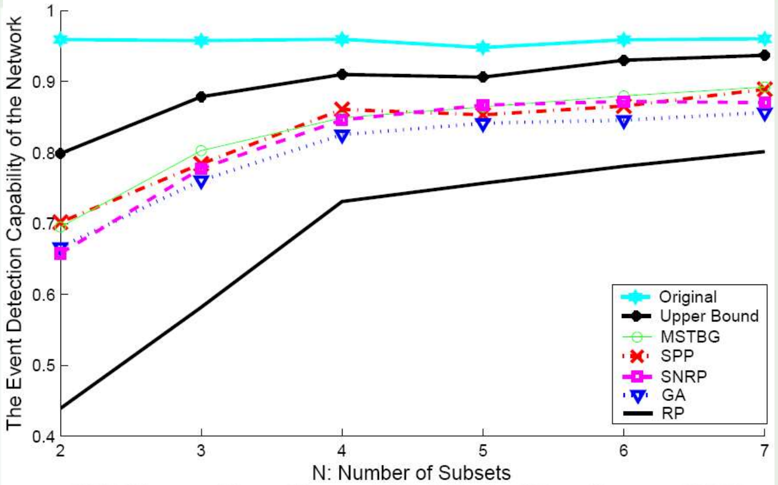
MSTBG tries to group nodes that are close to each other to different subsets so as to avoid the close-gathering of the nodes in the same subset.

Part 2: Simulation Studies

- The "Original" curve shows the EDC of the entire network when no subset is off
  - P_{all}
- The "Upper Bound" curve shows the EDC upper-bound of the network when one subset ceases to work
  - $1 - (1 - P_{all})^{\frac{N-1}{N}}$
  - This is a non-achievable upper-bound as it considers the non-achievable but optimum case where each subset has equal event detection probability at any point of the network
- RP, GA, SPP, MSTBG
- SNRP: Sensor Network Reconfiguration Protocol, a distributed implementation of MSTBG

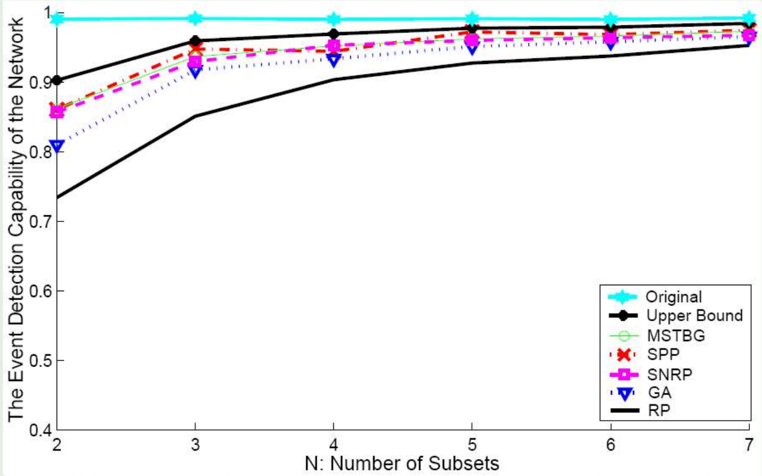
Part 2: Simulation Studies

node number = 100



EDC as a function of N (Node Number = 100)

node number = 150



EDC as a function of N (Node Number = 150)

- The naive RP algorithm performs by far the worst
- SPP, MSTBG, and SNRP always perform better then GA
- When $N > 4$, improving N cannot effectively improve EDC
  - Larger N incurs longer time for the entire network to complete a migration task

Part 2 Contributions

- We identify a critical problem faced by event-detection WSN when they are conducting migration tasks.
- We address this problem with several fast algorithms

Geographic Forwarding

A Waypoint-Based Geographic Forwarding Protocol for Data Collection WSNs

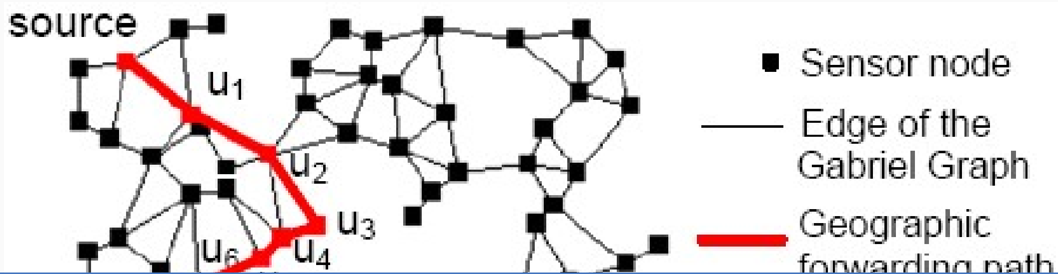
Part 3: Background

Geographic Forwarding

- Greedy forwarding
  - A node finds out its nearest neighbor to the sink
  - If the neighbor is nearer to the sink than the node, forward packet via it

Part 3: Geographic Forwarding Illustration

Geographic Forwarding



The detour-mode forwarding tends to forward data packets along the boundaries of holes.

→ The path from the source to the destination is much longer than the optimum.

→ More energy consumption in data collection.

Part 3: Topological Length

Definition

The topological length of a path is the total number of hops between the source and the destination of the path

Energy consumption in data forwarding

Part 3: Waypoints-based Geographic Forwarding

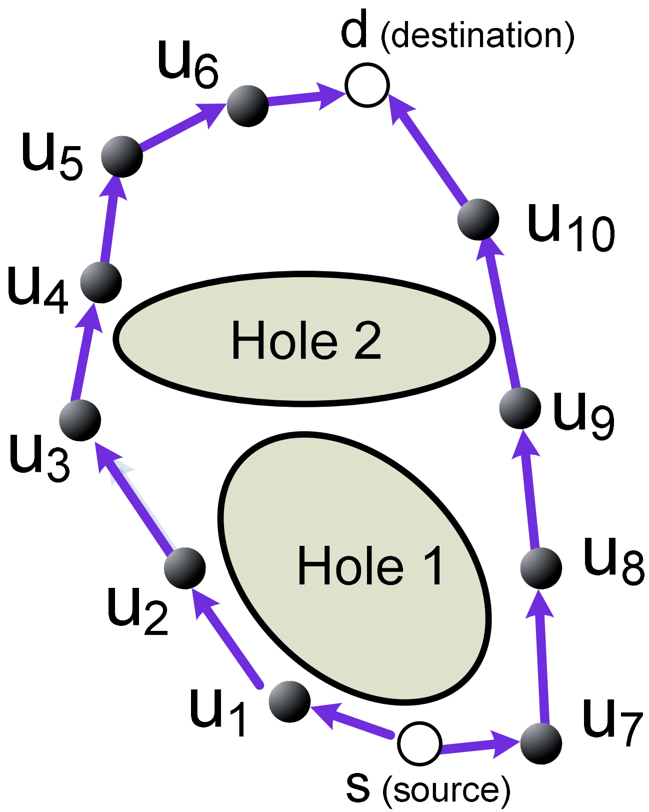
Waypoint-based geographic forwarding

- Find a waypoint sequence [w(1), w(2), ..., w(M)] where w(1) is the source and w(M) the destination
- Forward packets via the waypoint sequence one by one
  - Packets are transported between two adjacent waypoints with a geographic forwarding scheme.

Minimize the unnecessary detours so that the path can bypass holes and barriers.

Waypoints: Calculated with a trial-and-error approach

Part 3: Motivations



(a). A network scenario where greedy forwarding results in suboptimal path.

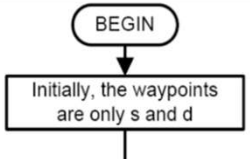
Existing schemes may result in suboptimal paths

Part 3: Geographic Data Reporting Protocol (GDRP)

Waypoint-Based Geographic Forwarding

Procedure 1 The mechanism of GDRP

$EndFlag \Leftarrow false$
$L_0 \Leftarrow \emptyset$
$j \Leftarrow 0$
$W_1 \Leftarrow [s, d]$
repeat

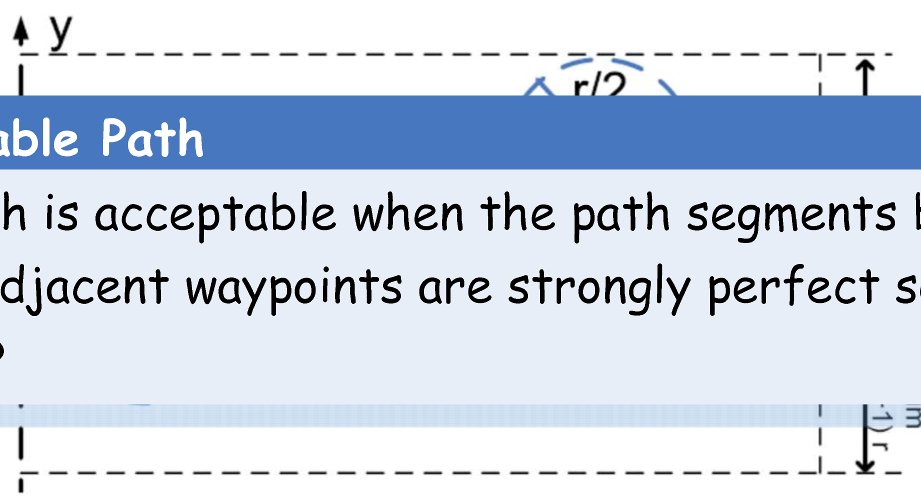


Waypoint-Based Geographic Forwarding

Part 3: Acceptable Path

strongly perfect sequence

- It is a greedy-forwarding path from the first node to the last node
- Given an x-y coordinate system with its x-axis passing the first and the last nodes, the maximum difference of the y-coordinates between any two nodes in the sequence is no more than $d = a \cdot r$, where a is a constant and r is the communication range.



Acceptable Path

- A path is acceptable when the path segments between two adjacent waypoints are strongly perfect sequence
- Why?

Part 3: Acceptable Path

Lemma

The topological length of a path is linearly related to the Euclidean distance between the source and the destination if the path is a strongly perfect sequence

Corollary

If an algorithm finds a path which is a strongly perfect sequence, the algorithm is a linear approximation to the shortest path algorithm in terms of the topological length

Part 3: Waypoint Calculation

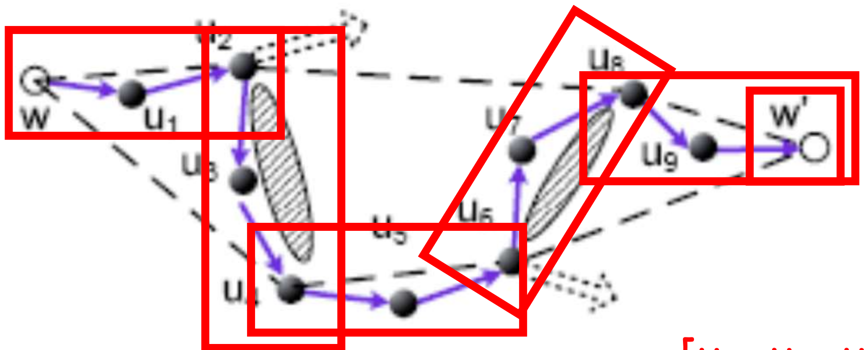
When a path is not an acceptable path

- There are holes or barriers in between the source and the destination
- At least one path segment between a pair of adjacent waypoints is not a strongly perfect sequence
- The impact of holes or barriers can be modeled as how they make the path segment "imperfect"
  - Find which parts of the segment make it fail to be a strongly perfect sequence

Part 3: Waypoint Calculation

Perfect sequence

- Similar to strongly perfect sequence
- With only one difference: It is not necessary that the last second node is connected with the last node.
  - It is not necessary to be a “path”



perfect sequences:

[w, u_1, u_2, w′]

[u_4, u_5, u_6, w′]

[u_8, u_9, w′]

[u_2, u_3, u_4] and [u_6, u_7, u_8] make the whole path

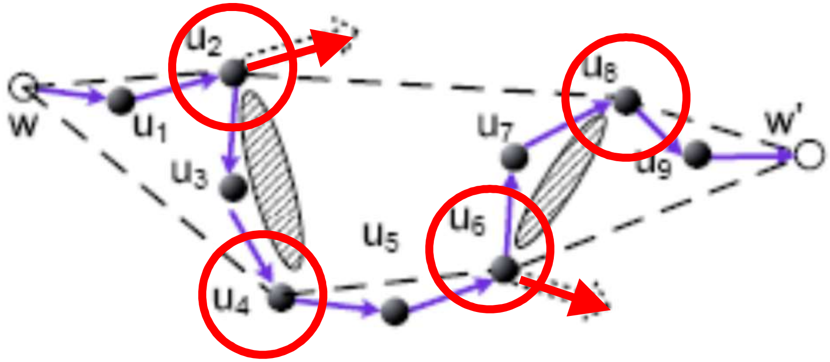
ect sequence

Holes and barriers

- Call these parts detour parts
- Save these parts: they are the current knowledge on the in-network holes and barriers

Part 3: Waypoint Calculation

Problem 1: Who should be the potential waypoints



1. The last node in a detour part should be a potential waypoint

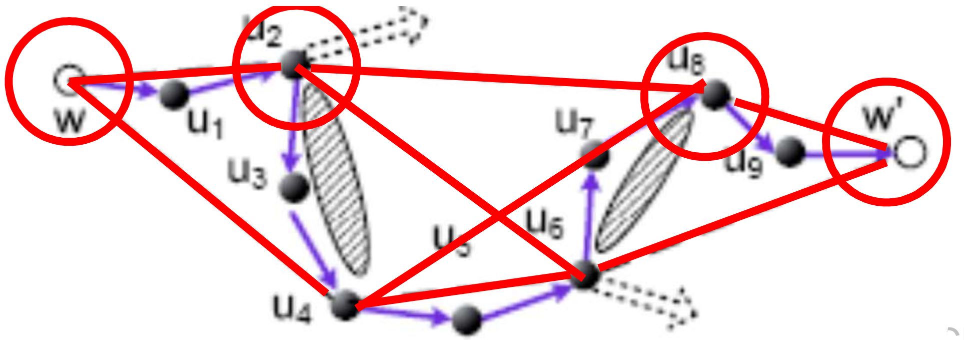
Reason: The hole or barrier does not influence the path any longer from the node on

2. The first node in a detour part should be a potential waypoint

Reason: The node can avoid the detour part by forwarding packets to another direction

Part 3: Waypoint Calculation

- Now we have a set of the potential waypoints, how to select a waypoint sequence for the next round?
- We expect in the next round, we can find an acceptable path



Hence, the waypoint sequence for the next round is [w, u_2, u_8, w′]

Part 3: Geographic Routing between Waypoints

Routing between adjacent waypoints

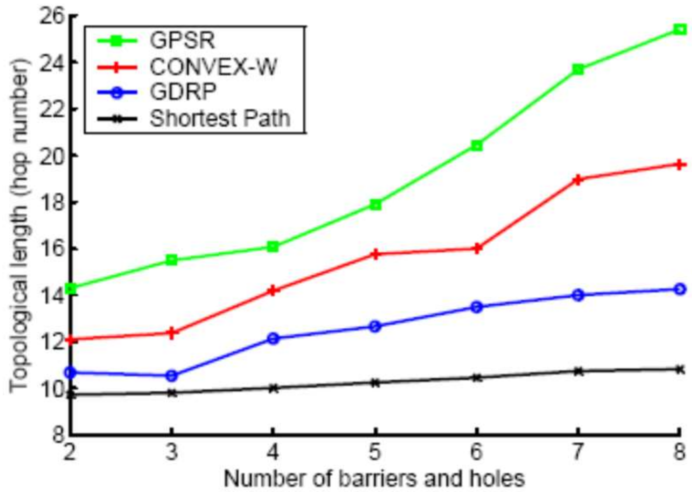
- Geographic forwarding
  - Detour-mode forwarding: routing along the face of the planar graph counter-clockwise or clockwise
  - Use one as default. Change if required by a waypoint (when it is a starting node of a detour part

Part 3: Simulation Studies

Protocols in simulation study

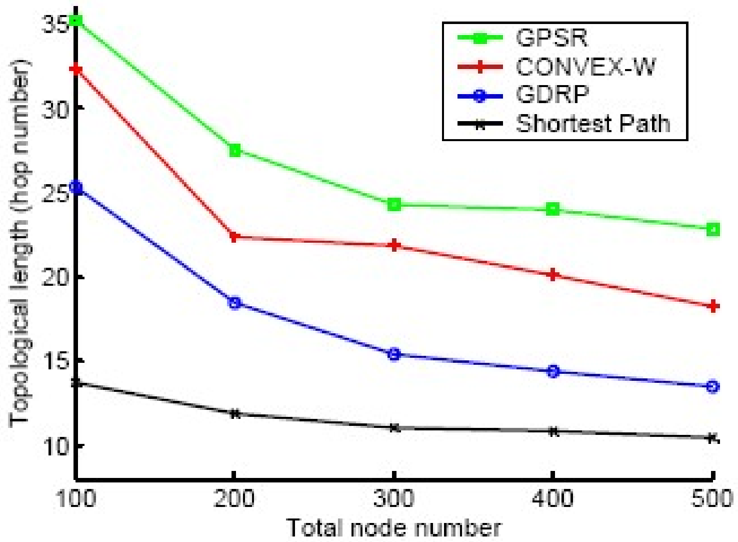
- GDRP (Geographic Data Reporting Protocol): Our protocol
- GPSR (Greedy Perimeter Stateless Routing): Traditional geographic forwarding
- CONVEX-W
  - Waypoint-based geographic forwarding
  - Packets are forwarded along the convex hull of the known holes

Part 3: Simulation Studies

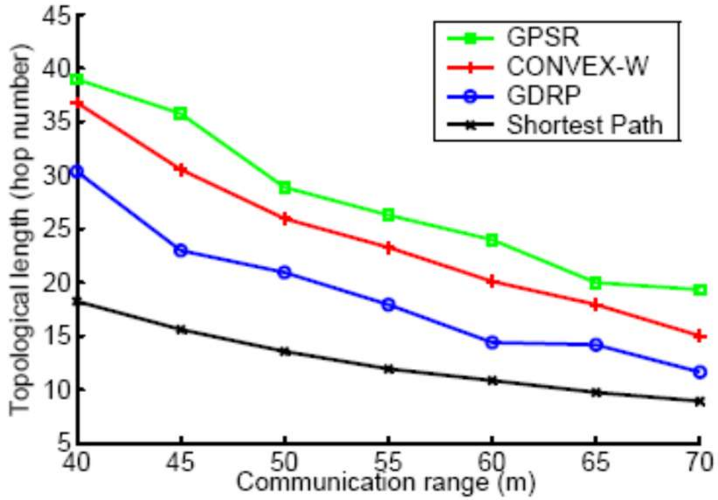


Topological lengths with different numbers of holes and barriers

Part 3: Simulation Studies



Topological lengths as a function of in-network node number



Topological lengths as a function of communication range

Part 3 Contributions

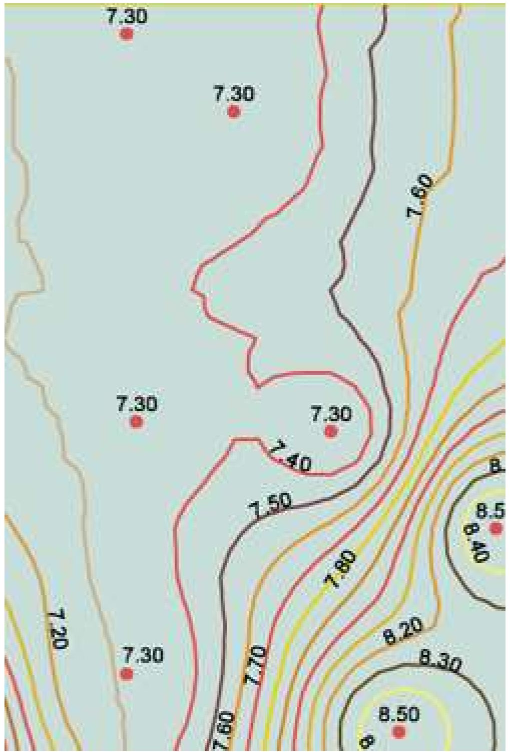
- We propose a waypoint-based geographic forwarding protocol
- We prove the performance guarantee of our protocol

An Energy-efficient Contour Mapping Service for WSNs

Part 4: Background

Background

- Data collection WSN
  - reconstruct the information of a scalar field of interest
  - The temperature distribution throughout a monitored space
  - The boundary where the concentration of a toxic gas reaches a dangerous level



Part 4: Motivation

Motivation: Current WSN contour mapping approaches cannot adaptively handle diverse user requests

- Optimized for either contour line or contour map enquiries, but not for both
  - Map: a set of line enquiries
  - Line: a map enquiry

⇒ Not energy efficient

Part 4: Online Active Contour Service (OACS)

Overview of OACS

- Divide the network into grids
  - Each grid has a head
    - in charge of the contour mapping computation
    - report the results to the sink
  - Purpose: in-network preprocessing for energy saving.
- Initially, only λ%'s nodes are on-duty, the rest are in sleep mode to save energy.

Part 4: OACS Overview

Overview of OACS

- Two kinds of enquiries are accommodated.
  - M-enquiries: contour map enquiry

    - parameter p: the proportion of sensor nodes that need to work to provide their readings
  - L-enquiries: contour line enquiry

    - parameter p

    - parameter cv: the value of the contour line requested
- The working sensors report readings to their corresponding heads
  - M-enquiry: all the working sensors report their readings
  - L-enquiry: those with reading close to cv report their readings

Part 4: OACS Overview

Overview of OACS

- The heads know the intensity of the field at a set of locations
  - Compute the contour line/map with kernel Support Vector Regression (SVR)
  - Check whether the precision requirement of the results matches p
    - yes, we are done!
    - no, select a best set of sleeping nodes, turn them on, obtain their readings, and refine the results

1. Why support vector regression
2. How to select a best set of nodes to open

Part 4: OACS Design Considerations

Design considerations

- Why kernel SVR
  - It equips OACS with a flexible method to deal with L-enquiries and M-enquiries
  - It can conveniently handle the nonlinear nature of contour lines.
  - The result of SVR is simple in representation.
    - a set of sensor readings/locations, and their weights
    - tens of bytes typically: a data packet is enough!

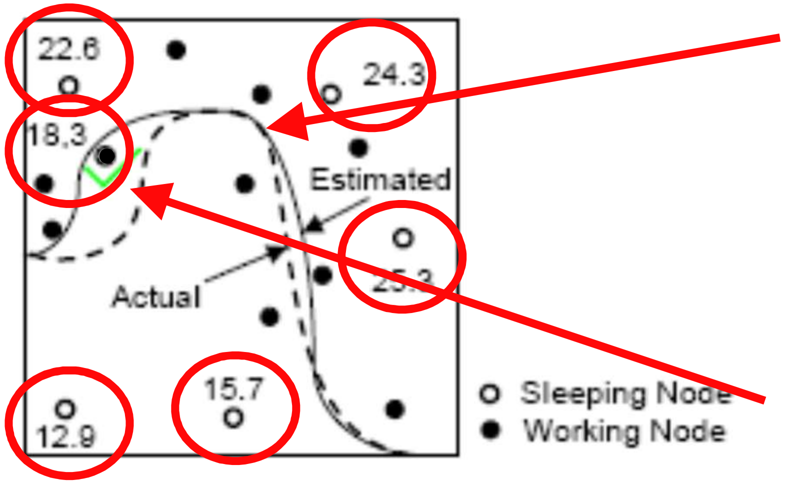
Precision tuning

- Initially only λ%'s nodes are working
  - If the precision requirement p is larger than λ, we should turn on another (p-λ)%'s nodes
  - How?

Part 4: OACS Design Considerations

Precision tuning details

- For an L-enquiry
  - Step 1: Regress the requested contour line based on the known sensor locations/readings
  - Step 2: Check whether the precision is acceptable
    - yes: We are done
    - no: Input the location of the sleeping nodes to the regressed function. Select the one with estimated reading the closest to the contour line, open it, and request its actual reading. Continue to step 1.



Regress the requested contour line (cv = 20) based on the known sensor locations/readings

Estimate the reading of the sleeping node based on the regression result

This sleeping node is the closest to the estimated contour line. So it is turned on to obtain its reading.

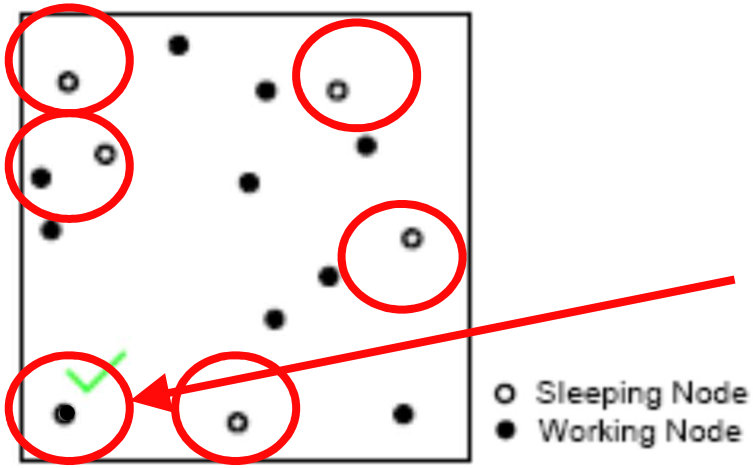
Part 4: OACS Design Considerations

Precision tuning details

- For an M-enquiry
  - Step 1: Regress the requested contour line based on the known sensor locations/readings
  - Step 2: Check whether the precision is acceptable

    - yes: We are done

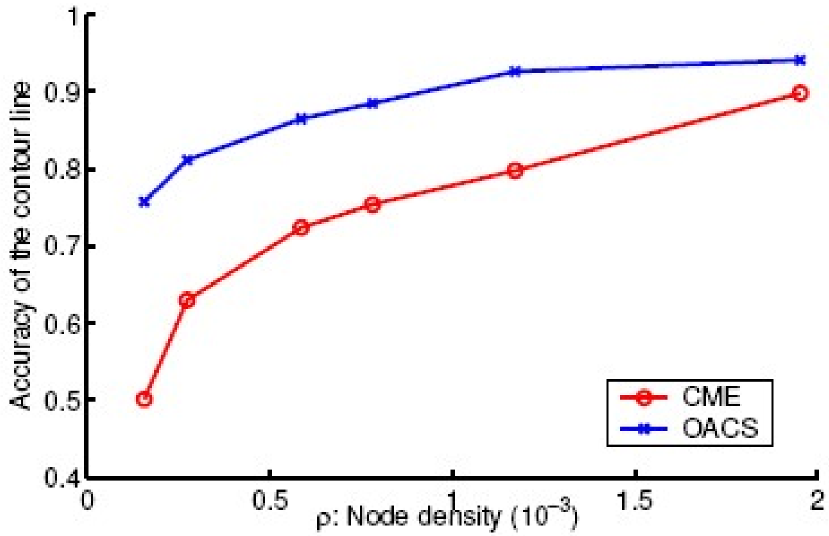
    - no: Select the sleeping node which is farthest to the working nodes, where such a distance is defined as the minimum distance between a sleeping node to any of the working nodes



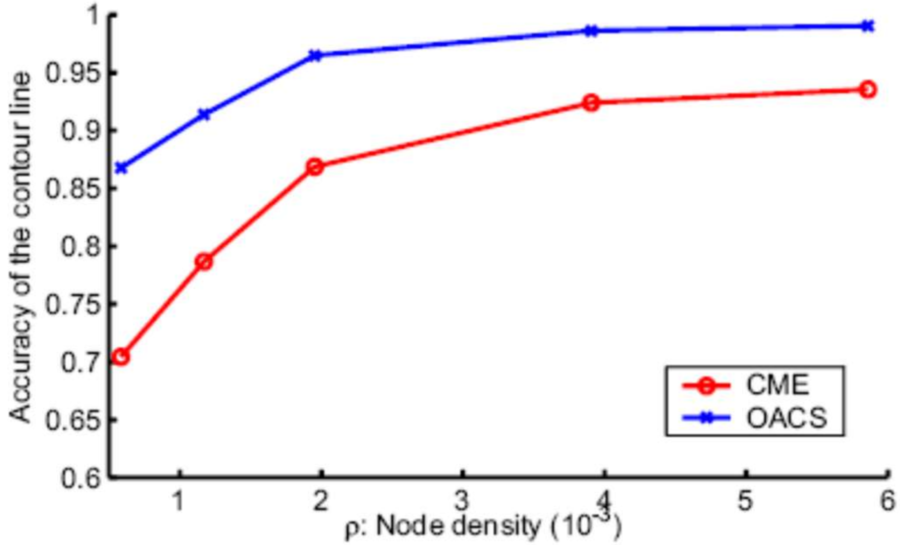
Check who is the farthest to the working nodes

This sleeping node is the closest to the estimated contour line. So it is turned on to obtain its reading.

Part 4: Simulation Studies



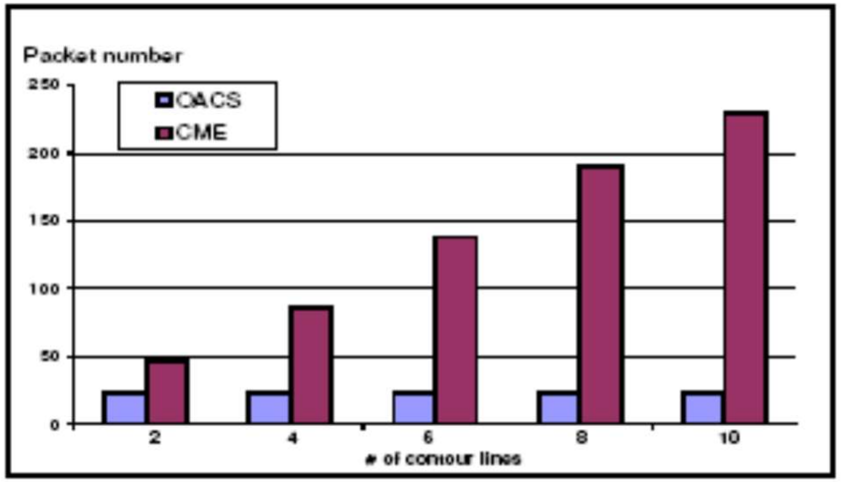
Accuracy comparisons between OACS and CME in processing L-enquiries



Accuracy comparisons between OACS and CME in processing M-enquiries

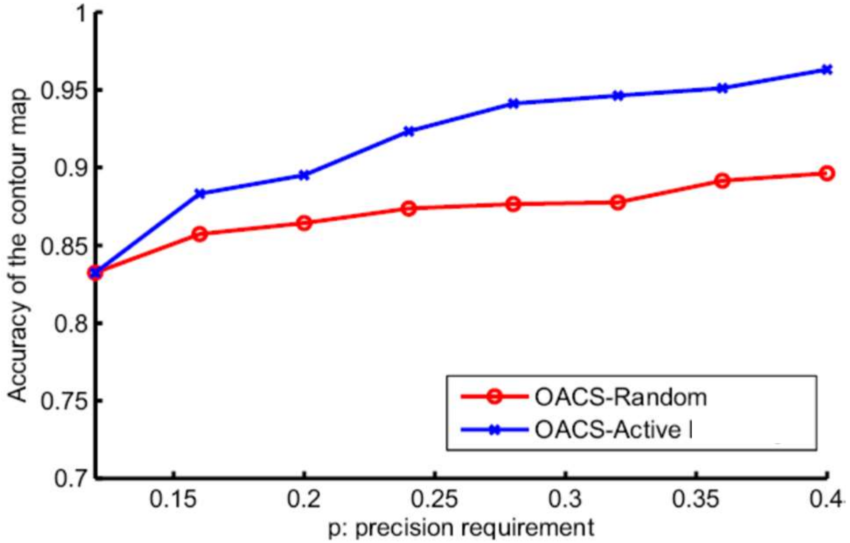
- Algorithm in comparison: Contour Mapping Engine (CME), the most up-to-date approach in WSN contour mapping

Contour Mapping

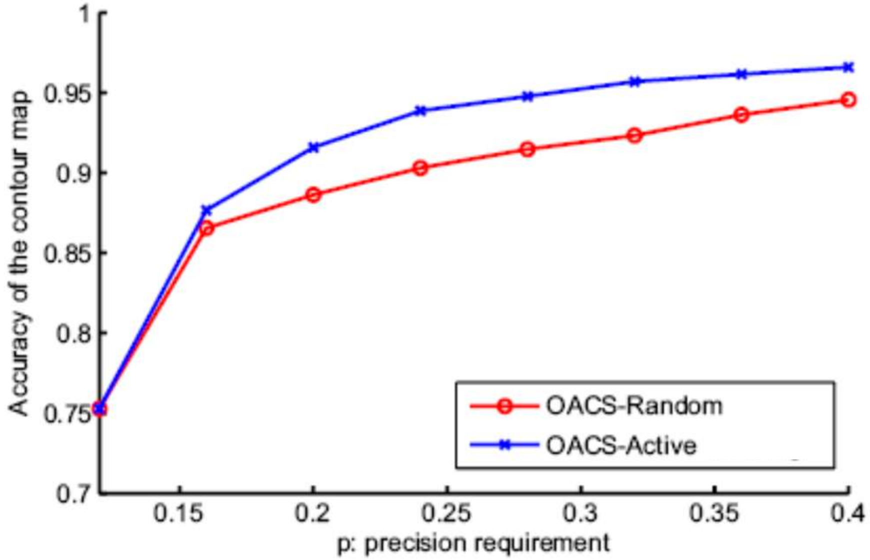


Total packet numbers sent by OACS and CME in processing M-enquiries

Contour Mapping



Accuracy as a function of precision requirement: L-enquiry case



Accuracy as a function of precision requirement: M-enquiry case

Part 4 Contributions

- We propose an contour mapping service that can handle diverse user requests energy-efficiently

Conclusion

- We address two critical coverage-oriented network partitioning problems for event detection WSNs. The algorithms are all fast algorithms with satisfactory performance
- We propose a waypoint-based geographic routing protocol for data collection WSNs which can find path with performance guarantee
- We propose a contour service that can provide a holistic energy-efficient treatments to diverse user requests for data collection WSNs

Event Detection →

Design requirements

- Maintain certain event detection probability
  - Coverage-oriented problems

Part 1: Reduce redundancy
Part 2: Maximize coverage

Data Collection →

Design requirements

- Reduce the forwarding packet number
  - Location-centric approaches

Part 3: Optimize the forwarding path
Part 4: Preprocess data in network

Appendix: Research Papers

Research Papers

Journal Publications

[1] Edith C.-H. Ngai, Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. A Delay-Aware Reliable Event Reporting Framework for Wireless Sensor-Actuator Networks, under minor revision, Ad Hoc Networks.

[2] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. A Fan-out Index and Its Application to Coverage-oriented Partitioning in Wireless Sensor Networks, Wireless Communications and Mobile Computing (To appear)

[3] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. On Sensor Network Reconfiguration for Downtime-Free System Migration, ACM/Springer Mobile Networks and Applications (MONET), Vol. 14, No. 2, pp. 241-252, April, 2009.

Research Papers

Conference Publications (Regular Research Papers)

[4] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. Surviving Holes and Barriers in Geographic Data Reporting for Wireless Sensor Networks, MASS'09

[5] Yangfan Zhou, Junjie Xiong, Michael R. Lyu, Jiangchuan Liu, and Kam-Wing Ng. Energy-efficient On-demand Contour Service for Wireless Sensor Networks, MASS'09

[6] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. On Sensor Network Reconfiguration Problem for Downtime-Free System Migrations, QShine'08

[7] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. An Index-Based Sensor-Grouping Mechanism for Field Coverage Wireless Sensor Networks, ICC'08

[8] Edith C.-H. Ngai, Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. LOFT: A Latency-Oriented Fault Tolerant Transport Protocol for Wireless Sensor-Actuator Networks, GlobeCom'07

[9] Yangfan Zhou, Edith C.-H. Ngai, Michael R. Lyu, and Jiangchuan Liu. POWER-SPEED: A Power-Controlled Real-Time Data Transport Protocol for Wireless Sensor-Actuator Networks, WCNC'07

[10] Yangfan Zhou and Michael R. Lyu. An Energy-Efficient Mechanism for Self-Monitoring Sensor Web, Aerospace'07

Research Papers

Earlier Publications

48 citations in total

[11] Edith. C.-H. Ngai, Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. Reliable Reporting of Delay-Sensitive Events in Wireless Sensor-Actuator Networks, MASS'06

[12] Yangfan Zhou, Haixuan Yang, Michael R. Lyu, and Edith C.-H. Ngai. A Point-Distribution Index And Its Application to Sensor Grouping in Wireless Sensor Networks, IWCMC'06

[13] Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu. On Setting up Energy-Efficient Paths with Transmitter Power Control in Wireless Sensor Networks, MASS'05

[14] Yangfan Zhou, Michael R. Lyu, Jiangchuan Liu, and Hui Wang. PORT: A Price-Oriented Reliable Transport Protocol for Wireless Sensor Networks, ISSRE'05

> 20 citations

Thank you!!!

Q & A