Delay-Oriented Reliable Communication and Coordination in Wireless Sensor–Actuator Networks

Ph.D. Thesis Defense
13 Jun 2007

Presented by Edith C.-H. Ngai

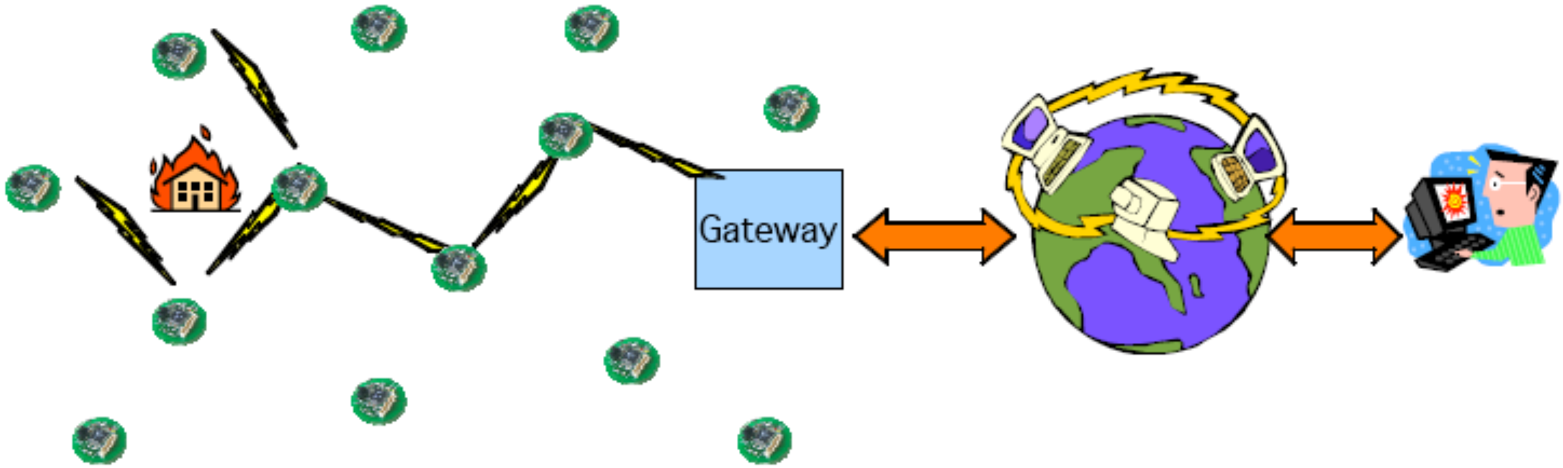
Supervisor: Michael R. Lyu

Outline

- Introduction
- Related Work
- Part 1: Real-Time Communication Framework in WSAN
- Part 2: Delay-Oriented Reliable Communication
  - 2.1 Delay-Aware Reliable Event Reporting
  - 2.2 Latency-Oriented Fault Tolerant Transport Protocol (LOFT)
  - 2.3 Power-Controlled Real-Time Transport Protocol (POWER-SPEED)
- Part 3: Delay-Oriented Reliable Actuator Coordination
  - 3.1 The Route Design Problem (RDP)
  - 3.2 Adaptive Delay-Minimized Route Design (PROUD)
- Part 4: Intruder Detection Against Sinkhole Attack
- Conclusion and Future Direction

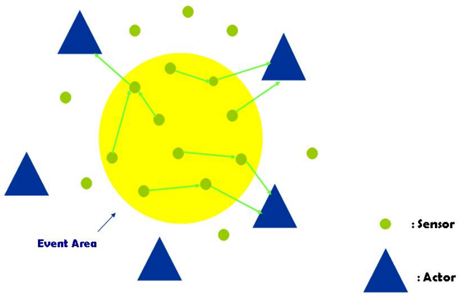
WSN

- Distributed and large-scale like the Internet
- A group of static sensors
  - resource constrained
  - wireless communications



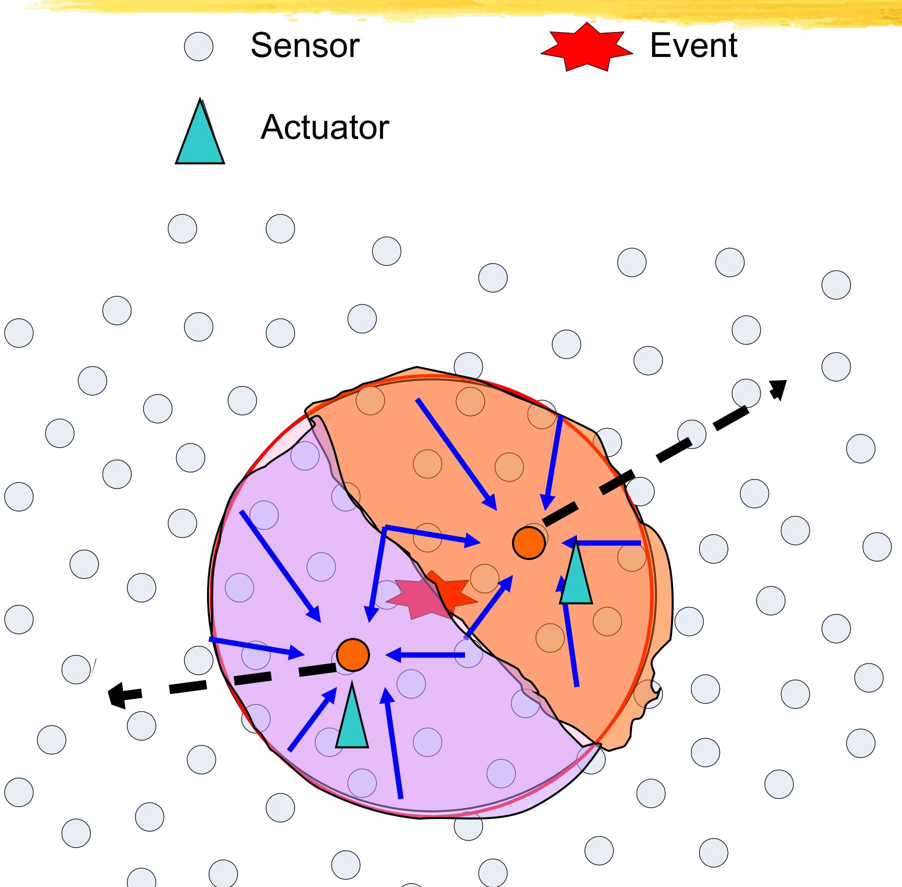
WSAN

- Collection of sensors and actuators
- Sensors
  - numerous resource-limited and static devices
  - monitor the physical world
- Actuators
  - resource-rich devices equipped with more energy, stronger computation power, longer transmission range, and usually mobile
  - make decisions and actuate adaptively in response to the sensor measurements



Applications

Operations in WSAN



(1) Event Happens

(2) Data Aggregation

(3) Event Reporting

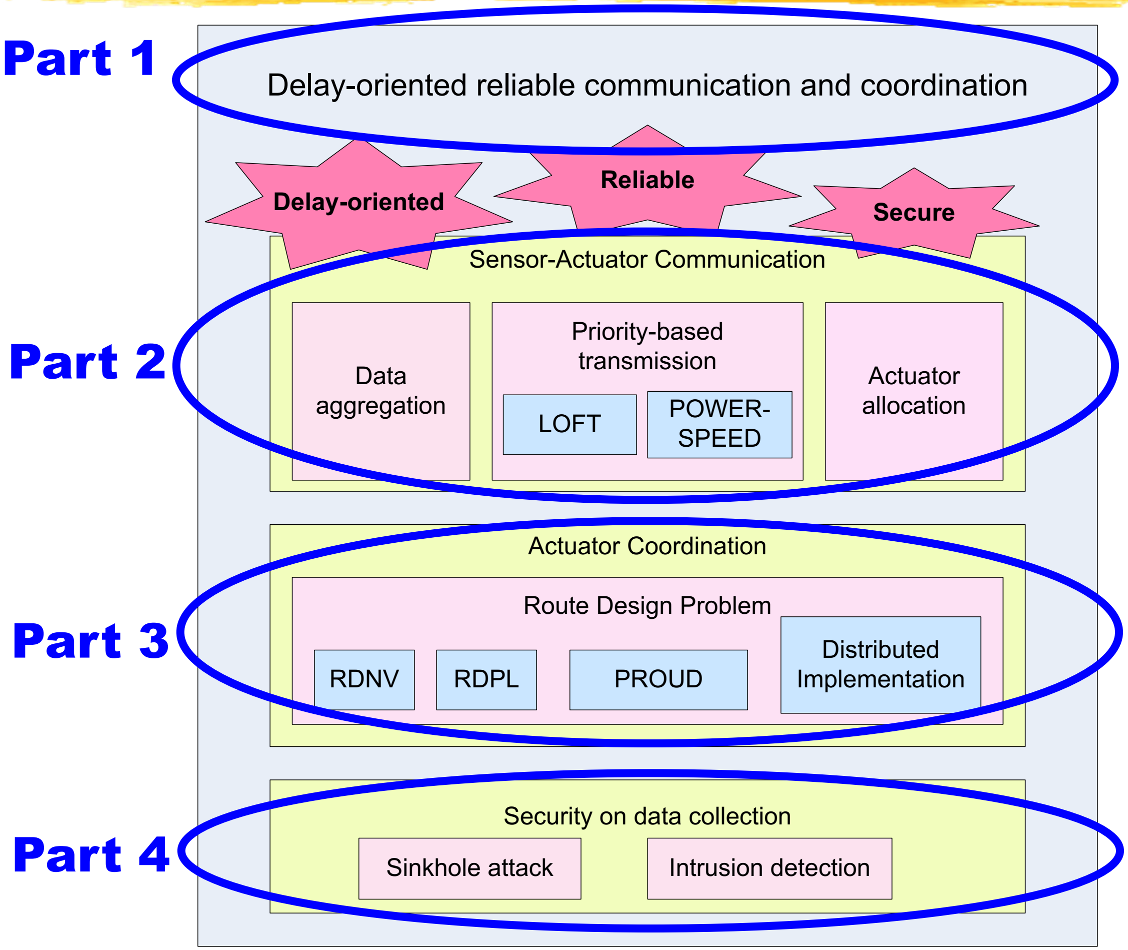
(4) Response to the Event

Event-driven applications

Self-organized, distributed, fast response

Thesis Focus

- Delay-oriented reliable communication
  - Real-time event reporting
  - Resist to node/link failures
- Delay-oriented coordination among actuators
  - Cooperative route design
  - Non-uniform sensor distribution
- Security issue
  - Intrusion detection in WSN



Contribution

- We studied real-time communication and coordination in WSAN
- We proposed a general reliability-centric framework for delay-aware event reporting for WSAN
- We considered delay, reliability, fault-tolerant, and energy efficiency in data transport
- We presented the Route Design Problem and proposed effective schemes to coordinate the actuators and minimize the data collection time
- We proposed an adaptive delay-minimized route design algorithm which can handle network dynamics
- We studied the security issues in sensor networks and proposed an efficient intruder identification algorithm against Sinkhole attacks

Real-time Communication Protocol in WSN

- SPEED [Hu et al. 2003]
  - Combines feedback control and non-deterministic QoS-aware geographic forwarding
- Velocity Monotonic Scheduling [Lu et al. 2002]
  - Packet scheduling policy that accounts for both time and distance constraints
- MMSPEED [Felemban et al. 2005]
  - Multi-path and multi-speed routing protocol for probabilistic QoS guarantee in WSN

Reliable Transmission with Error-prone Sensors

- Node-level fault tolerance (NLFT) [Aidemark et al. 2005]
  - Masks transient faults locally by using time-redundant task scheduling in nodes
- Bi-criteria scheduling heuristic [Assayad et al. 2004]
  - Uses heuristic in data-flow graph to maximize reliability and minimize runtime
- Routing in DTN [Jain et al. 2005]
  - Applies erasure code and data replication

Heterogeneous Sensor Networks

- Anycast communication paradigm [Hu et al. 2004]
  - Constructs an anycast tree rooted at each event source and updates the tree dynamically
- Power-aware many-to-many routing [Cayirci et al. 2005]
  - Actuator broadcasts registration messages, while sensors build their own routing tables
- Distributed coordination framework [Melodia et al. 2005]
  - Sensors forward readings to the appropriate actuators by the data aggregation trees

Mobile Elements

- Architecture using moving entities (Data Mules) to collect sensor data [Shah et. al. SNPA'03]
- Mobile sinks with predictable and controllable moving pattern [Chakrabarti et al. IPSN'03, Kansal et al. Mobisys'04]
- Mobile sinks can find the optimal time schedule to stay at appropriate sojourn points [Wang et al. HICC'05]
- Message ferry (MF) approach to address the network partition problem in sparse ad hoc network [Zhao et al. Mobihoc'04]

Mobile Elements (cont.)

- Joint mobility and routing algorithm with mobile relays to prolong the network lifetime [Luo et al. Infocom'05]
- Partitioning-based algorithm to schedule the movement of mobile element (ME) to avoid buffer overflow and reduce min. required ME speed [Gu et al. Secon'05]
- Vehicle routing problem (VRP)
  - Considers scheduling vehicles stationed at a central facility to support customers with known demands
  - Minimize the total distance traveled

Intrusion Detection in WSN

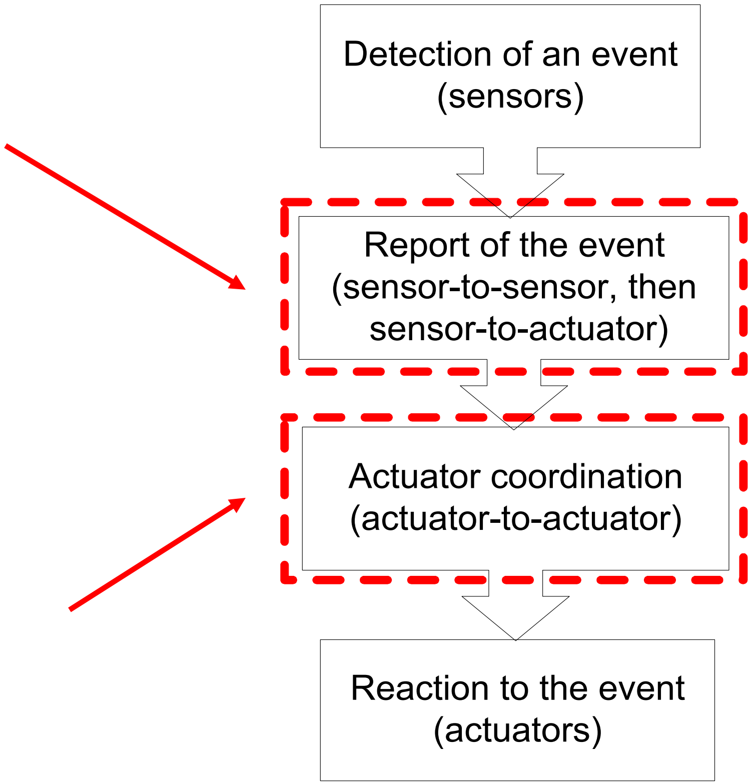
- Mechanism for detecting and mapping jammed regions [Wood et al. RTSS'03]
- Algorithm for the identification of faulty sensors and detection of the reach of events [Ding et al.]
- Trace the identities of the failed nodes with the topology conveyed to the base station [Staddon et al. WSNA'02]
- A Statistical En-route Filtering (SEF) mechanism that can detect and drop false reports [Ye et al. Quality and Reliability Engineering Int.'01]
- A packet leash mechanism for detecting and defending against wormhole attacks [Hu et al. Infocom'05]

A Real-Time Communication Framework for Wireless Sensor-Actuator Networks

Part 1

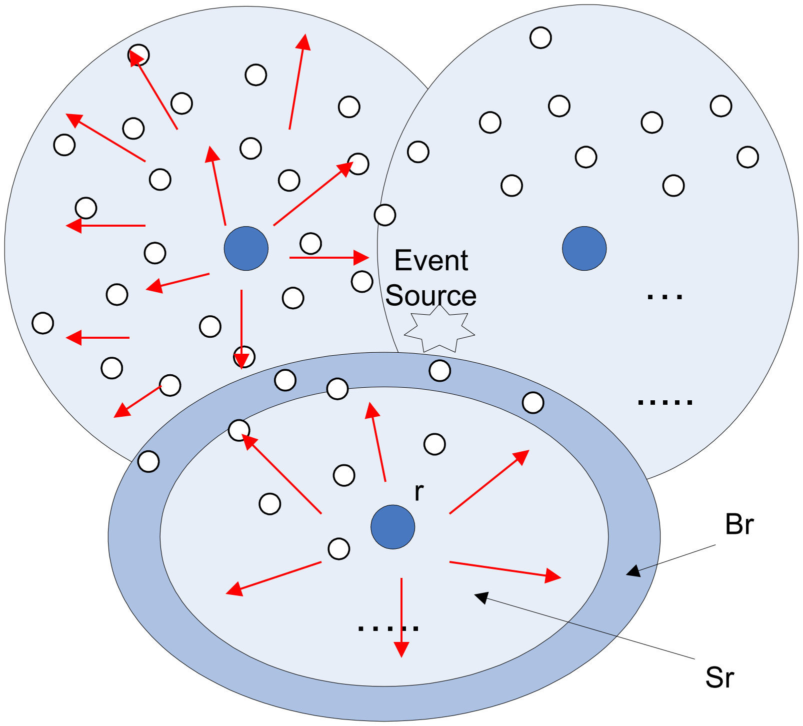
A Real-time Communication Framework for WSN

- Event reporting
  - Detection of an event
  - Formation of map and data aggregation
  - Data transmission
- Actuator coordination
  - Combination of maps
  - Location update



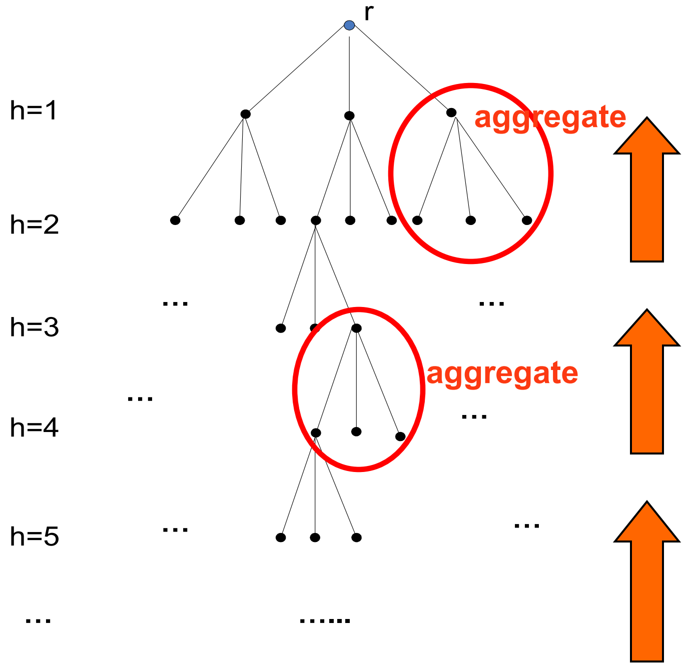
Formation of Maps

- To reduce the network traffic, the sensor will aggregate event reports and perform data fusion from the neighboring nodes
- The sensors *r*, which detected an event the earliest, start the formation of maps



Data Aggregation

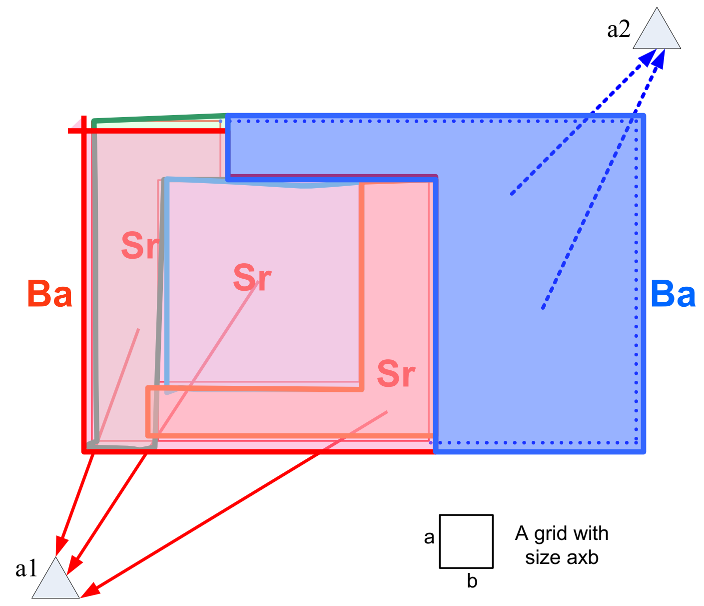
- When a node receives the replies from its descendent nodes, it concatenates its own reply and forwards them to the previous hop
- Nodes with even number of depth *h* concatenate the reply with its own coordinates and sensed data
- Nodes with odd number of depth *h* aggregate the data from their immediate descendents before forwarding them



Combination of Maps

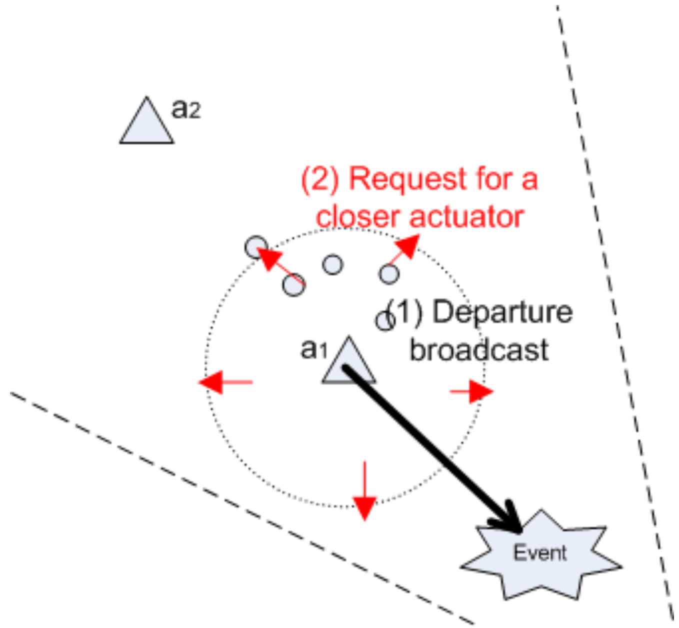
Algorithm Combination of Maps

for each actuator *a* on event *e*,
  if (received multiple S_r)
    Gather the B_r in grid coordinates from all S_r
    Remove the redundant B_r
    Remove the connected B_r
    Store the remaining B_r in B_a
  end if
  Exchange the B_a with other actuators
  Remove the redundant B_a
  Remove the connected B_a
  Estimate the B_a by finding lower-left and upper- right grids $<x_{min}, y_{min}>$ *and* $<x_{max}, y_{max}>$
end for



Location Update

- Update the location of actuator to sensors
- Plan the optimal location of the actuators for efficient reactions



Summary of Part 1

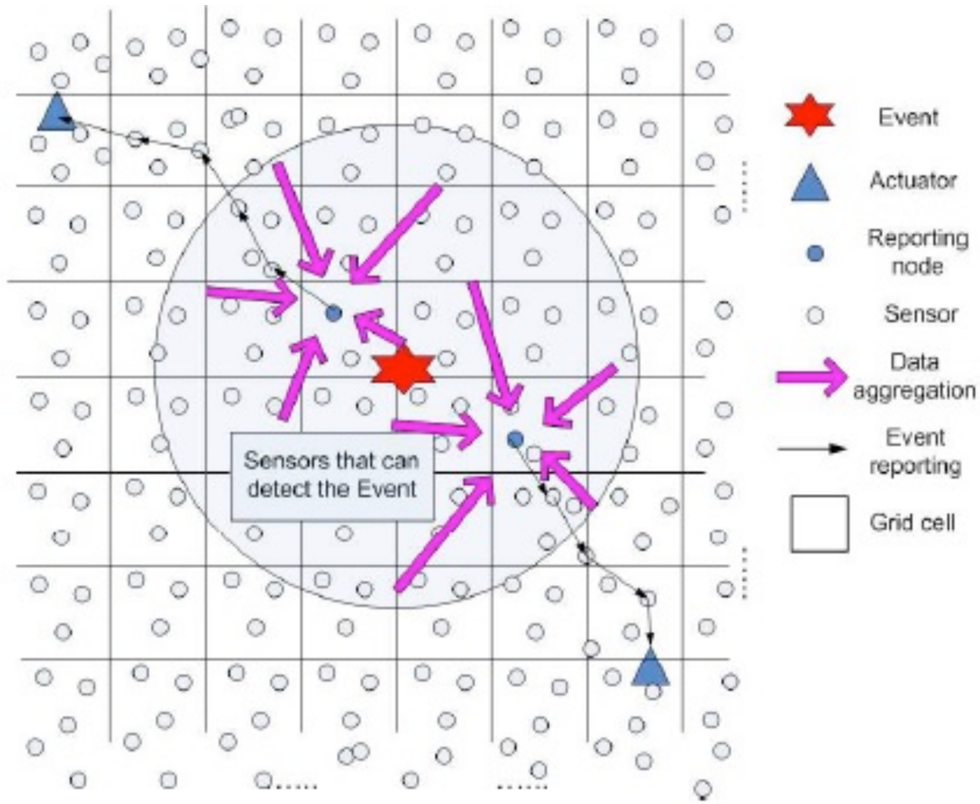
- A real-time communication framework for WSAN is presented
- It provides an efficient data aggregation algorithm that reduces network traffic
- It considers layered data transmission to minimize the delay
- It provides an actuator coordination algorithm with combination of maps for effective reaction
- It offers a distributed, self-organized, and comprehensive solution for real-time event reporting and reaction for WSAN

Delay-Oriented Reliable Communication in Wireless Sensor-Actuator Networks

Part 2

Network Model

- Compose of sensors and actuators
- Nodes aware of their locations
- Divide the network into a number of grid cells for data aggregation
- A subset of nodes, referred as reporting nodes, send data to the actuators
- Anycast routing



Objective

- Reliability index
  - Measures the probability that event data are aggregated and received accurately within pre-defined latency bounds

System Parameters

e : Event

q_e : Data report of event e

Q_e : Set of data reports of event e that satisfy the end-to-end latency constraint

$Imp(e)$: Importance of event e

B_e: Latency bound for sensor-actuator reporting of event e

D_{q_e}: End-to-end delay of data report q_e

N_e: Number of data reports for event e

f: Probability of failures in data aggregation

Objective

Maximize

$$\mathbb{R} = \sum_{\forall e} Imp(e) * r_e, \quad (1)$$

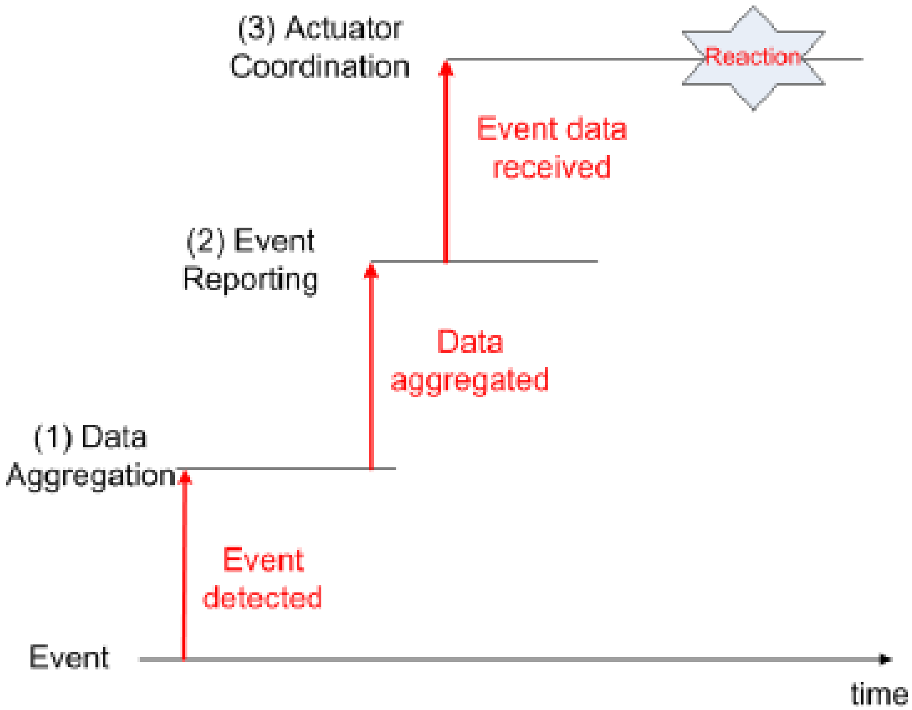
where $r_e = \frac{|Q_e|(1-f)}{N_e}$.

Subject to

$$D_{q_e} \leq B_e \quad (2)$$

Workflow of Framework

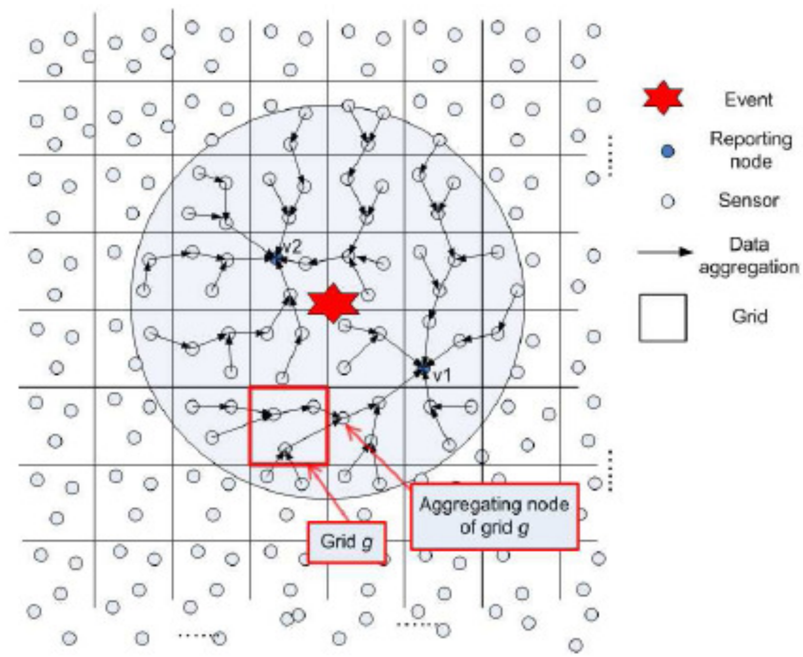
1. A multi-level data aggregation scheme, which is fault-tolerant with error-prone sensors
2. A priority-based transmission protocol, which accounts for both the importance and delay requirements of the events
3. An actuator allocation algorithm, which smartly distributes the actuators to match the demands from the sensors.



Grid-Based Data Aggregation

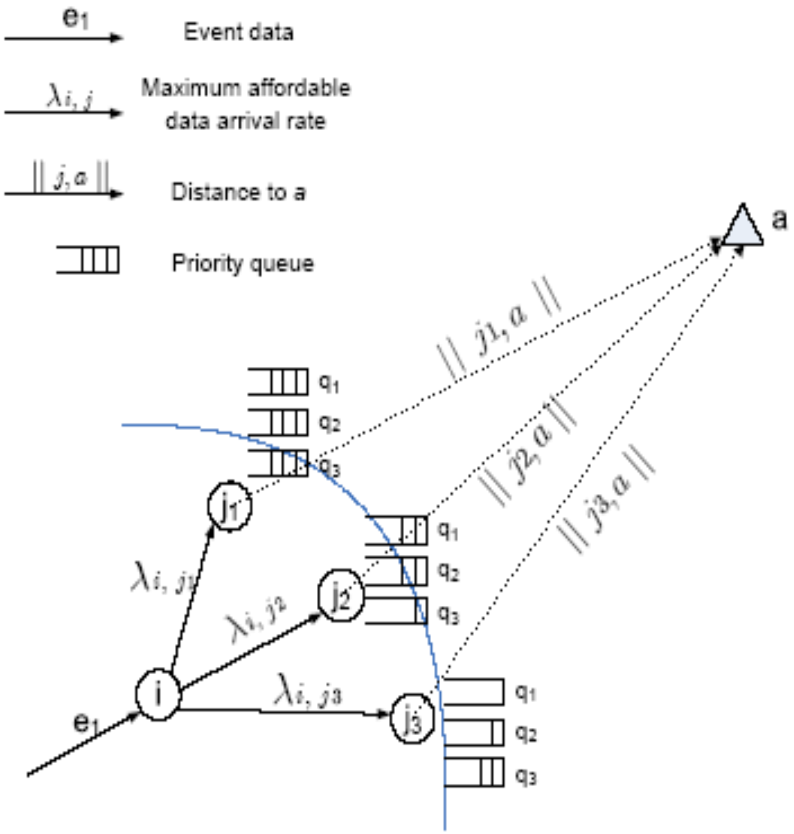
Algorithm 1 Data Aggregation

Define: $\overline{x_g}$ as aggregated data mean of grid g;
for each sensor s receive data x_i **do**
  if multiple $x_i \in g$ and s is the aggregating node **then**
    find the median med among data $<x_1, x_2, ..., x_n>$;
    for each data $x_i \in g$ **do**
      if x_i - $med > \Delta d$ **then**
        blacklist node i
      end if
    end for
    $\overline{x_g}$ = mean of the un-blacklisted data $x_i \in g$
  end if
end for



Priority-Based Event Reporting

- Priority queues
  - prioritized scheduling to speed up important event data transmission
  - queue utilization as an index for route selection to meet the latency bounds
  - first-in-first-out (FIFO) discipline



Queueing Delay

- The queueing delay of the highest priority queue: $\overline{d_{q1}} = \overline{R} + \overline{S}\,\overline{N_q^1}$, where

$\overline{R} = \frac{1}{2}\sum_{k=0}^{K}\lambda_k\overline{S^2}$ is the mean residual service time in the node, $\overline{N_q^1}$ is the mean number of packets in first queue, K is the number of priority queues, λ_k is the arrival rates of the packets in priority queue k, $\overline{S}$ and $\overline{S^2}$ are the expectation and second moment of the service time of the sensor.

- The queueing delay of *k*

$$\overline{d_{q_k}} = \frac{\overline{R}}{(1-\rho_1-\ldots-\rho_{k-1})(1-\rho_1-\ldots-\rho_k)}$$

Next Hop Selection

- Consider node *i* receives new type of event data $data_e$
- It broadcasts a control message to its immediate neighbors
- Every neighbors *j* replies with the message: $< a, \overline{S}, \lambda_{high}, \lambda_{low} >$,

where a is the target actuator, $\overline{S}$ is the expected service time of node j,

$\lambda_{high} = \sum_{\forall k, imp(data_k) \geq imp(data_e)} \lambda_k$ is the sum of all λ_k with data equal or more important than $data_e$,

$\lambda_{low} = \sum_{\forall k, imp(data_k) < imp(data_e)} \lambda_k$ is the sum of all λ_k with data less important than $data_e$.

Next Hop Selection

- The end-to-end delay to actuator should be less than the latency bound B_e
- Node *i* first estimates the advancement $h_{i,j}$ towards the actuator *a* from *i* to *j*, and then the maximum delay from *i* to *j*, $delay_{i,j}$.

$$h_{i,j} = \frac{\| a, i \| - \| a, j \|}{\| a, i \|}$$

So,

$$delay_{i,j} \leq B_e * h_{i,j}$$

Since $delay_{i,j} = d_q + d_{tran} + d_{prop} + d_{proc}$, the maximum queueing delay $d_{q_{max}}$ is:

$$d_{q_{max}} = B_e * h_{i,j} - (d_{tran} + d_{prop} + d_{proc})$$

Next Hop Selection

- Only neighbors with $dq_{max}>0$ will be considered as next hop
- Node *i* starts inspecting the neighbors with $\lambda_{high}=0$ and $\lambda_{low}=0$
  - $\lambda_{low}=0$ means it will not affect the transmission time for the existing packets in that node
  - $\lambda_{high}=0$ means it can be served with the highest priority
- Node *i* calculates the maximum data rate $\lambda_{i,j}$ that it can forward while satisfying the latency bound:

$$\rho_{i,j} < 1 - \lambda_{high}\overline{S} - \frac{\overline{R}}{(1-\lambda_{high}\overline{S})d_{q_{max}}},$$

where $\rho_{i,j} = \lambda_{i,j}\overline{S}$ is the maximum affordable load of j for handling data from i on event e.

- Data packets are forwarded to the neighbor with the highest $h_{i,j}$ and $\lambda_{i,j}$

Actuator Allocation

- The actuators may record the event frequency and re-arrange their standby positions periodically

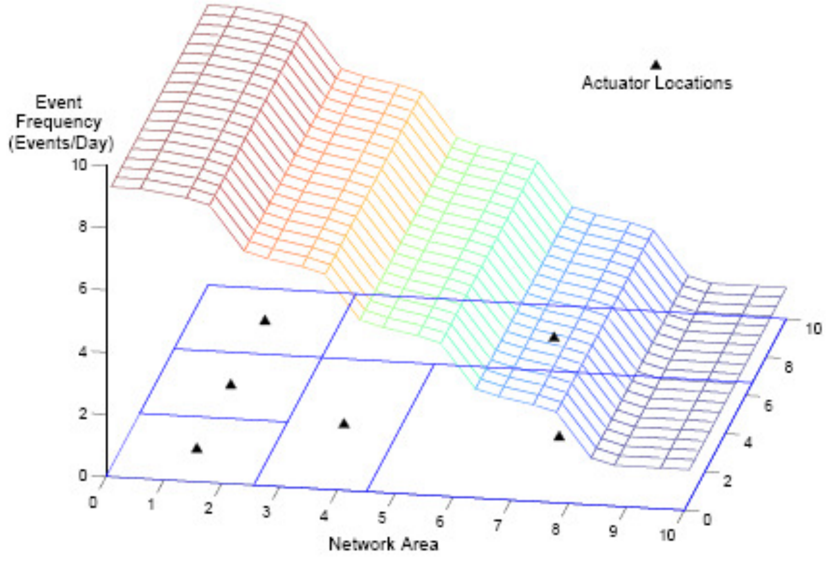


Fig. 4. Actuator Allocation with 6 Actuators.

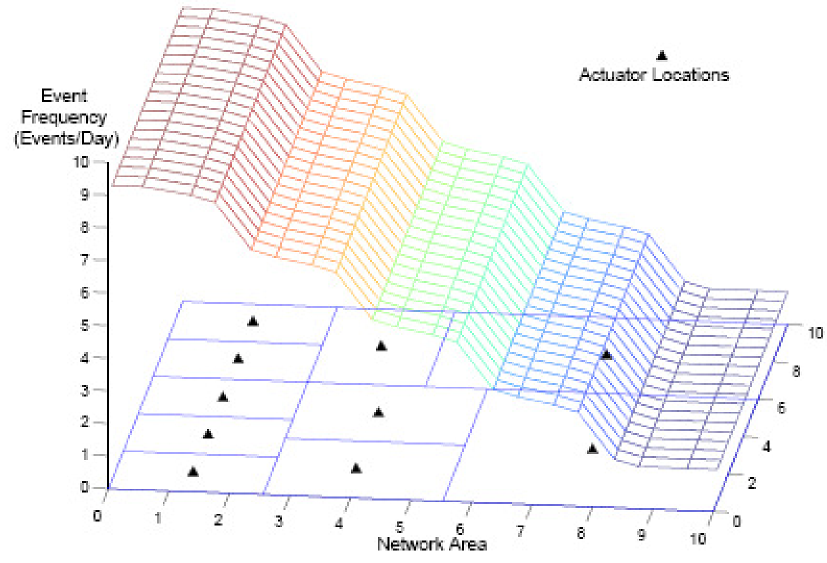


Fig. 5. Actuator Allocation with 10 Actuators.

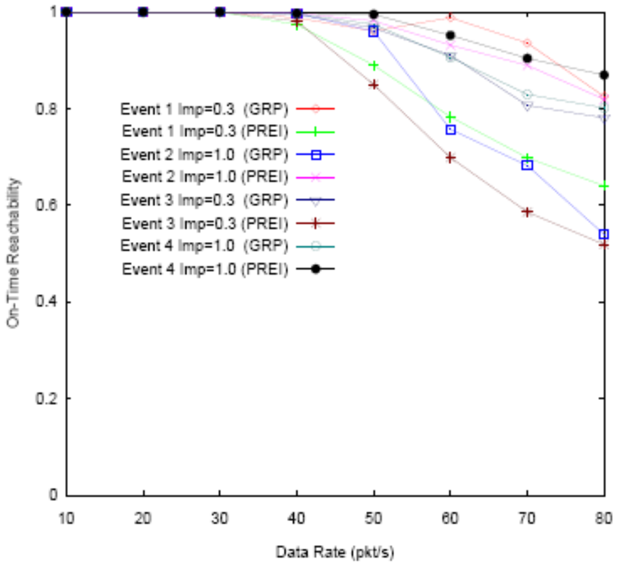
Simulations

- Simulator: NS-2
- Metrics
  - On-time Reachability
  - Average Delay
  - Overall Reliability
- 4 events
  - 2 with high importance
  - 2 with low importance
  - Located at left bottom corner

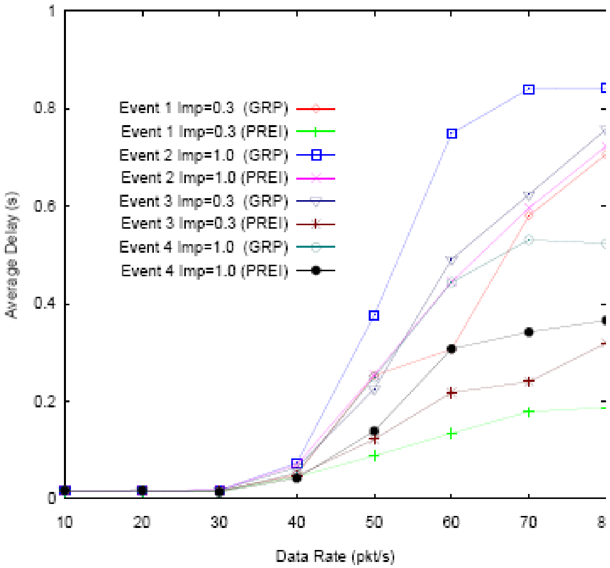
SIMULATION PARAMETERS

Network size	200m x 200m
No. of sensors	100
Node placement	Uniform
Radio range	40m
MAC layer	IEEE 802.11
Bandwidth	2Mbps
Packet size	32 bytes
No. of actuators	1-6
No. of concurrent events	3-10
B_e	2sec

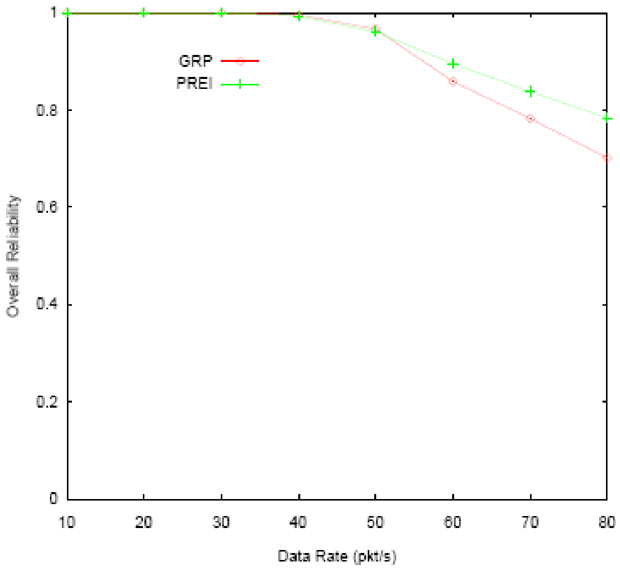
Reliability of Event Reporting



On-Time Reachability



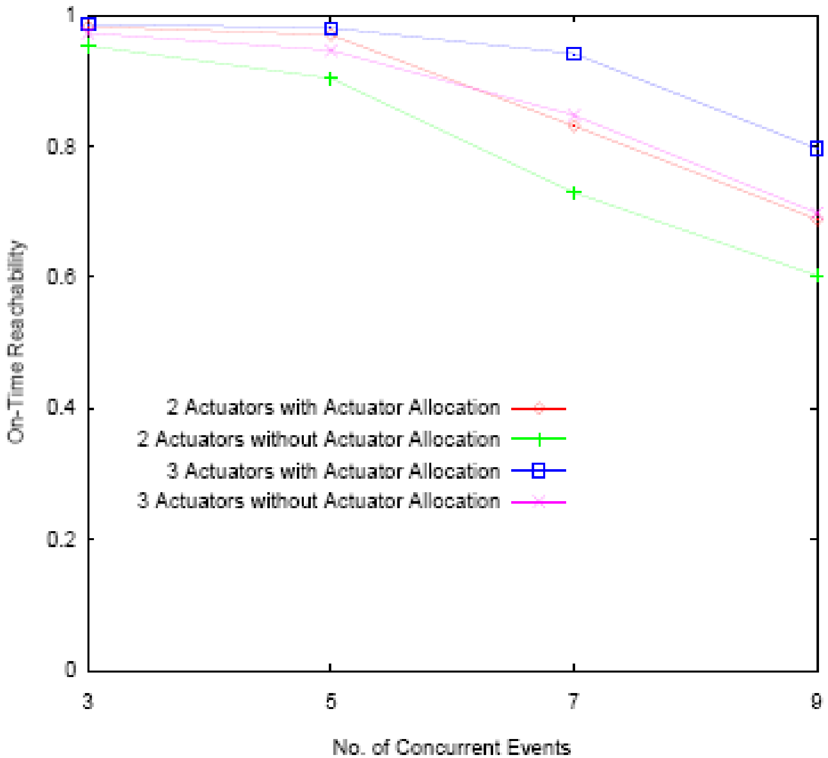
Average Delay

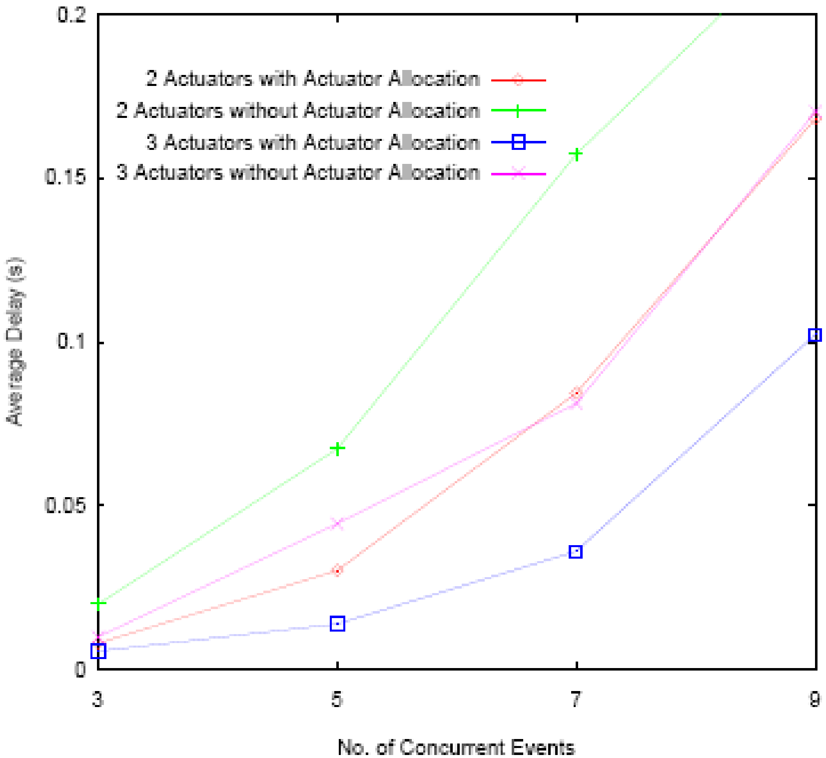


Overall Reliability

Actuator Allocation

- Divide whole field into three, with event occurrence probability 0.6, 0.333, and 0.067
- Data rate = 60pkt/s



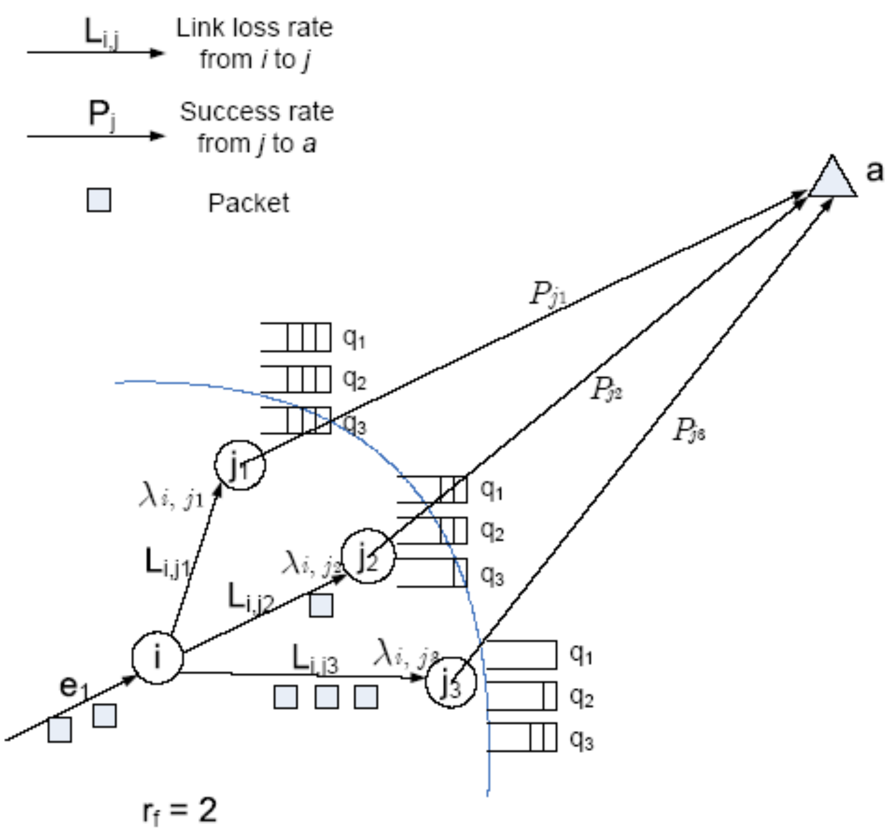


Latency-Oriented Fault Tolerant Transport Protocol (LOFT)

- Transmission errors, buffer overflow, node failures along the path
- Different levels of reliability can be obtained based on the requirements of various event data
- LOFT adopts adaptive packet replication to handle link failures and provide reliability on the success arrival of packets

Coping with Transmission Failures

- Consider node *i* and its potential next hops j_1, j_2, and j_3
- The allocation of packets from *i* to its neighbors is proportional to their maximum affordable arrival rates $\lambda_{i,j1}$, $\lambda_{i,j2}$, *and* $\lambda_{i,j3}$ *to* balance the load
- Node *i* may check if the estimated path success rate can meet the event reliability requirement R_{req}
- If not, it decides the replication factor r_f to meet the requirement and forwards the replicated packets to the next hops



Coping with Transmission Failures

- Node i selects the top k neighbors with the highest $h_{i,j}$ and satisfactory $\lambda_{i,j}$ and estimates their link loss rates $L_{i,j}$
- Each neighbor j periodically reports the number of packets it received from node i, so that i can calculate the loss rate $L_{i,j}$ with the number of packets it sent to j in a particular time interval
- Then, it can obtain the link loss rate by an EWMA (Exponentially Weighted Moving Average) approach with its previous and current estimations of the link loss rate
- Then, i estimates the path success rate P_j from i to a via j as follow:

$$P_j = (1 - L_{i,j})^{1/h_{i,j}}$$

Coping with Transmission Failures

- Sensor *i* will allocate packets to its neighbors according to their $\lambda_{i,j}$
- The neighbors with higher $\lambda_{i,j}$ will be allocated with more code blocks. The proportion $prop_j$ of packets to neighbor *j* is:

$$prop_j = \frac{\lambda_{i,j}}{\sum_{n=1}^{k} \lambda_{i,n}}$$

- The probability that the packet can be delivered successfully from *i* to *a* by these *k* neighbors, P_i, can then be estimated as:

$$P_i = \sum_{j=1}^{k} \left(\frac{\lambda_{i,j}}{\sum_{n=1}^{k} \lambda_{i,n}} * P_j\right)$$

- Then, node *i* determines the replication factor r_f with the following equation:

$$r_f = ceil(R_{req}/P_i)$$

Coping with Transmission Failures

- Each node *j*, which received the packets, selects the next hop m' with the highest $h_{j,m}$ and satisfactory $\lambda_{j,m}$
- Similarly, the path success rate obtained must be greater than R_{req}:

$$(1 - L_{j,m'})^{1/h_{j,m'}} \geq R_{req}$$

- If the link loss rate from *j* to m' satisfies the above equation, packet will be forwarded to m'. Since the reliability of a path is composed by a series of links on it:

$$(1-\overline{L}_1)(1-\overline{L}_2)(1-\overline{L}_3)...(1-\overline{L}_n) > R_{req},$$

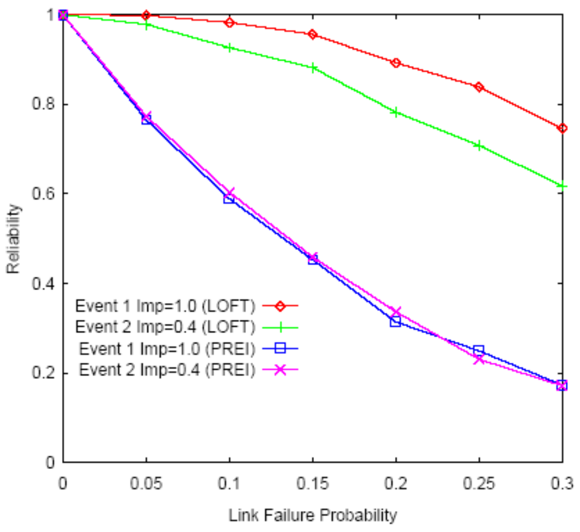
and

$$(1-\overline{L}_2)(1-\overline{L}_3)...(1-\overline{L}_n) > R_{req}/(1-\overline{L}_1)$$

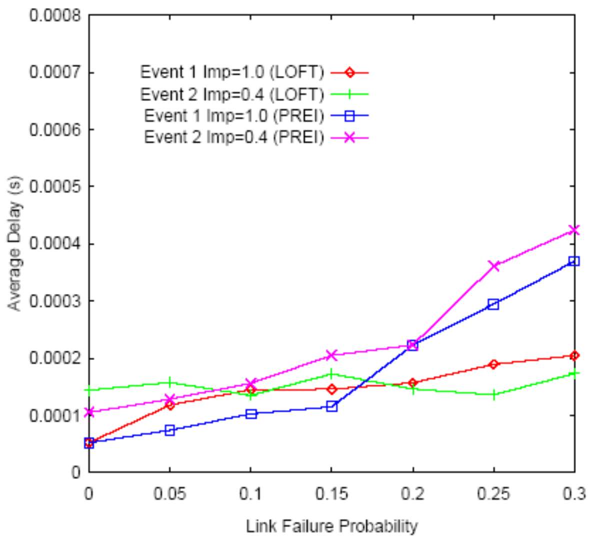
- Node *j* updates the reliability R_{req} and forwards it with the packets to the selected neighbor m':

$$R'_{req} = R_{req}/(1 - L_{j,m'})$$

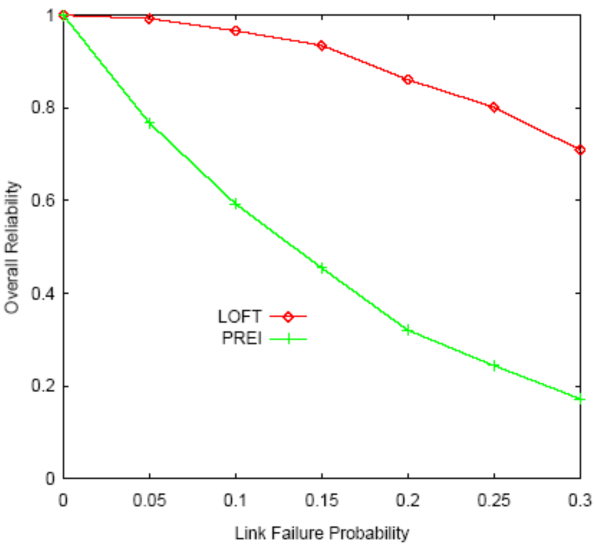
Varying Failure Rate (Data Rate = 15pkt/s)



Reliability

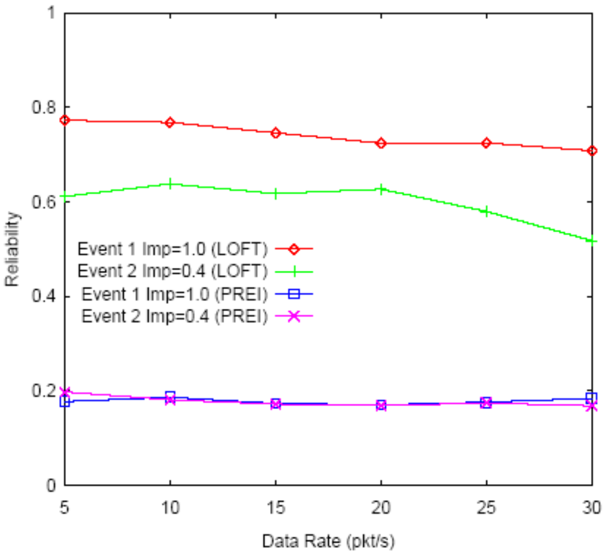


Average Delay

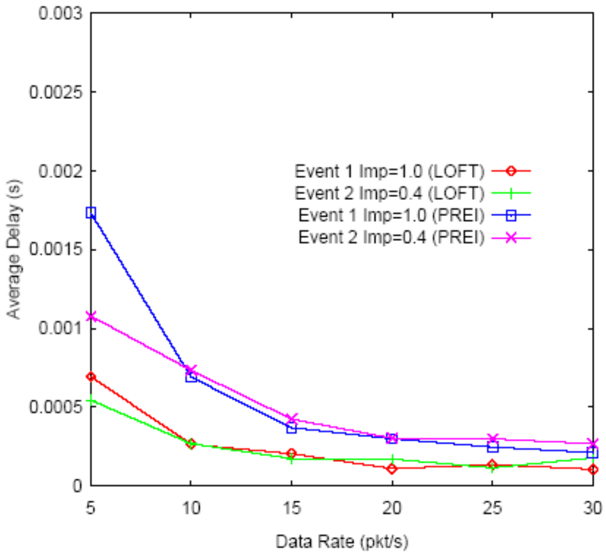


Overall Reliability

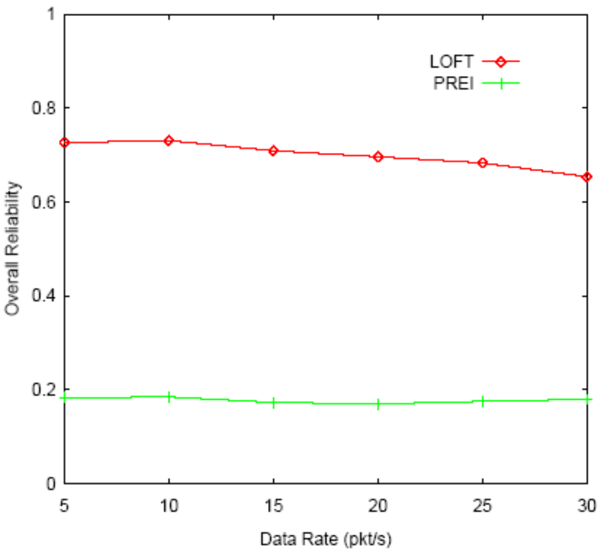
Varying Data Rate (Failure Rate = 0.3)



Reliability



Average Delay



Overall Reliability

POWER-SPEED

- A real-time and energy-efficient data transport protocol for WSAN
- Sensor nodes select the next-hop neighbor to actuators according to the spatio-temporal historic data of the upstream QoS condition,
- An adaptive transmitter power control scheme conveys packets in an energy-efficient manner
- Reduce energy consumption of data transport while ensuring the QoS requirement in timeliness domain

Select Next-hop Neighbor

- Node s_n should minimize the total energy consumption required for a packet to reach an actuator
- Since energy consumption for receiving and processing a packet is constant, we consider the energy required for sending a packet
- $E(s_n,w)$ denotes the energy for a packet to travel from s_n to w:

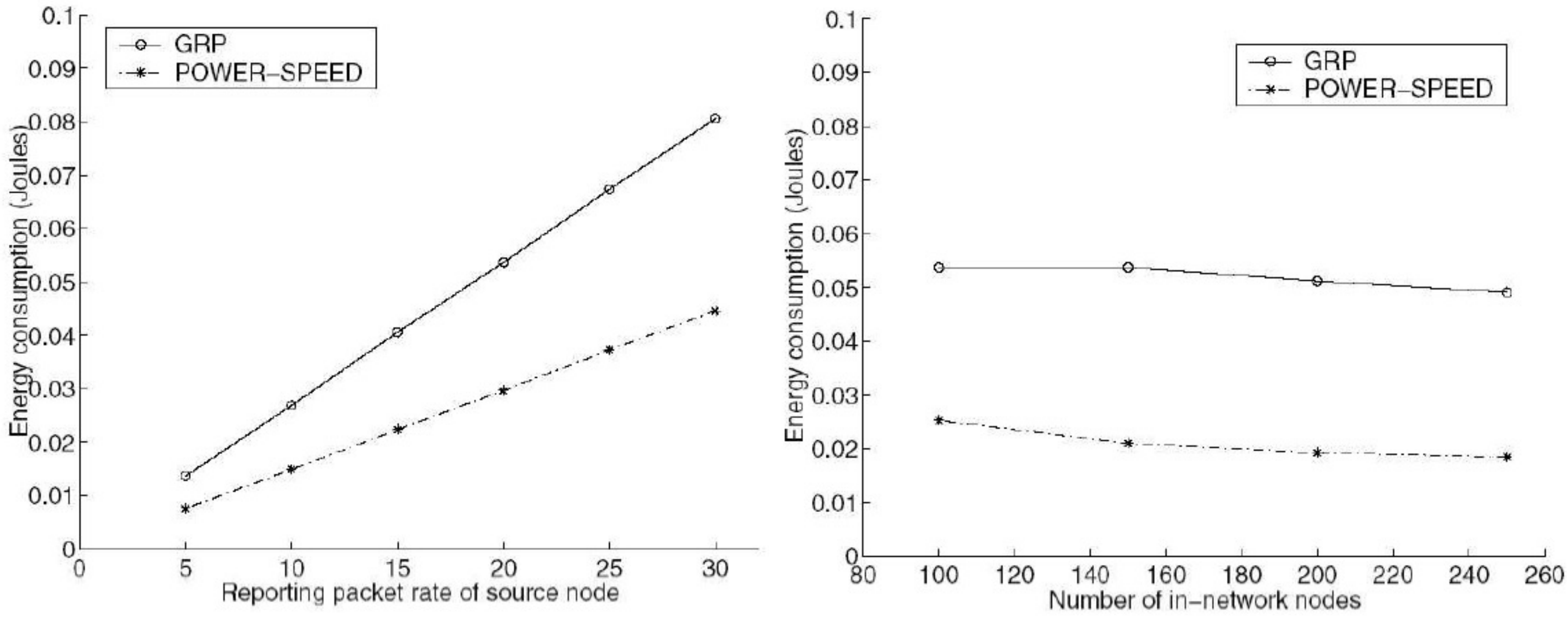
$$E(s_n, w) = \gamma Pr(s_n) = \gamma c \cdot (Dis(s_n, w))^n$$

- Node s_n estimates the total energy consumption for the packet to reach actuator a_j via node w by:

$$total\_e(s_n, w) = hops \cdot E(s_n, w)$$

- Node s_n selects the node that achieves the minimum value of total energy consumption, while satisfying the delay bound

Energy Consumption



Summary of Part 2

- We presented a distributed and comprehensive solution for reliable event reporting and actuator coordination in WSAN
  - We provided a distributed data aggregation mechanism
  - We proposed a reliable priority-based event reporting scheme
  - We proposed an actuator allocation algorithm
- We proposed LOFT for handling node/link failures
- We proposed POWER-SPEED for real-time energy efficient data transport
- Simulation results are provided to demonstrate the effectiveness of our solutions

Delay-Oriented Reliable Actuator Coordination for Wireless Sensor-Actuator Networks

Part 3

Motivation

- Given
  - Each static sensor has a limited buffer
  - Non-uniform data generation rates among the sensors
  - Sensor stores locally sensed data and uploads the data until some actuator approaches
- Strategy
  - Actuator visits locations with higher importance (i.e. higher data rate) more frequently
- Question
  - How to minimize the inter-arrival time from the actuator to the static sensors???

    => Route Design Problem (RDP)

System Parameter

s	Sensor node
R_s	Communication range of sensor node
c_{ij}	Cost from sensor location i to j
x_{ijk}	Boolean indicating whether link (i, j) is on route k
W_j	Weight of sensor location j (a value between 0.0 and 1.0)
N_j	Number of sensors with weight W_j
A_j	Average actuator inter-arrival time for sensor location j
T_k	Period of route k
N	Number of sensor locations
M	Number of actuators

Route Design Problem (RDP)

Route Design Problem (RDP):

$$Minimize \sum_{\forall j} A_j * W_j * N_j, \qquad (1)$$

where $T_k = \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ijk} * c_{ij}$, $A_j = \mathbb{F}(T_1, T_2, ..., T_{M'})$ is the function of all T_k that pass through the sensor location j.

Subject to:

$$\sum_{k=1}^{M} \sum_{i=1}^{N} x_{ijk} \geq 1, \forall j = 1, ..., N \qquad (2)$$

$$||s, j|| \leq R_s, \forall s, \exists j \qquad (3)$$

$$\sum_{i=1}^{N} x_{ipk} - \sum_{j=1}^{N} x_{pjk} = 0, \forall k = 1, ..., M, p = 1, ..., N \qquad (4)$$

$$y_j - y_i + N \sum_{k=1}^{M} x_{ijk} \leq N - 1, i \neq j = 1, ..., N \qquad (5)$$

$$x_{ijk} \in \{1, 0\}, \forall i, j, k; \; y_i \text{ arbitrary}, \qquad (6)$$

Definition and Property

Definition 1. A route is a tour walked through repeatly by an actuator.

Property 1. *Route with a Hamiltonian cycle achieves shorter maximum inter-arrival time* A_{max} *than that without.*

Property 2. *The average actuator inter-arrival time* A_j *of a sensor* j *on multiple routes can be calculated as* $\mathbb{F}(T_1, T_2, ..., T_{M'}) = \frac{\prod_{k=1}^{M'} T_k}{\sum_{k=1}^{M'}(\prod_{i=1, i \neq k}^{M'} T_i)}$.

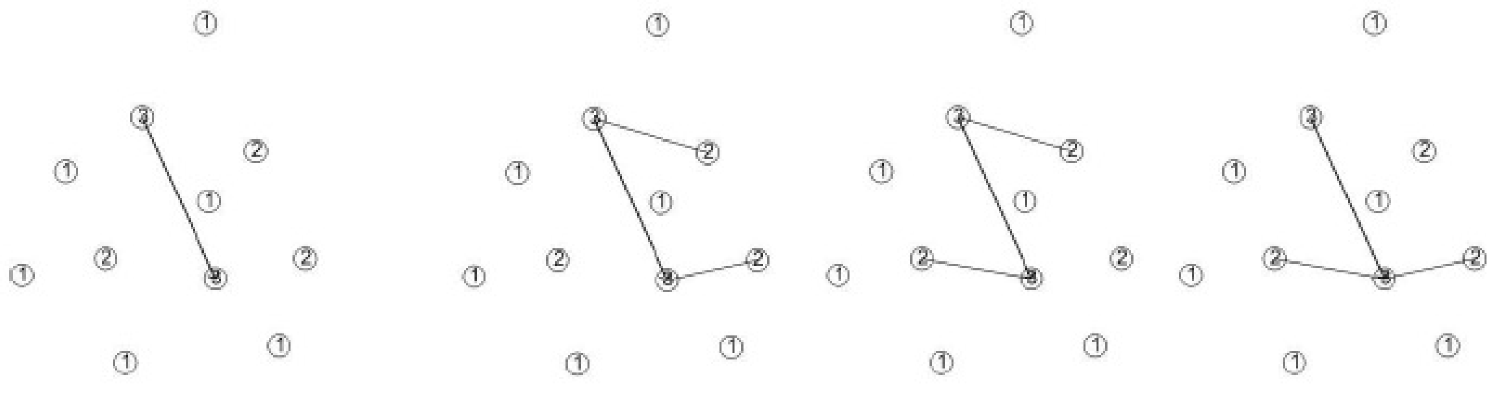
Theorem 1. *The route design problem (RDP) is NP-hard.*

Route Design Algorithm by Varying Number of Visits (RDNV)

- Design independent routes for multiple actuators
- Utilize multiple minimum spanning trees (MSTs)
- Construct M routes with equal period where highly weighted sensors will be visited more frequently
  - A sensor location with weight W_i will be visited by W_i *M actuators (routes)
  - E.g. W_i = 0.75, M=4 => N_i = 3
  - If all routes have the same period T, from property (2), the average inter-arrival time A_{avg} will be $T/3$

(1) Clustering with MSTs

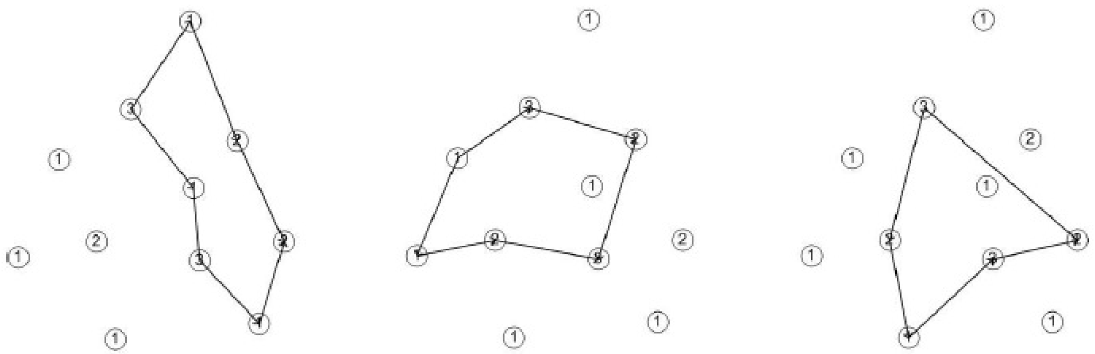
- N_i = ceil (W_i * M)
- Locations with the same N_i belong to the set S_i
- Our algorithm builds M spanning trees T_k, where $k = 1, \ldots, M$
  - Locations with highest $N_i = M$ will be included in all trees
  - Then, the locations with the next highest N_i will be assigned to N_i trees with lowest costs
  - The process repeats until there is no remaining locations



(1) T1, T2, T3 (2) T1 T2 T3

(2) Form a TSP Solution

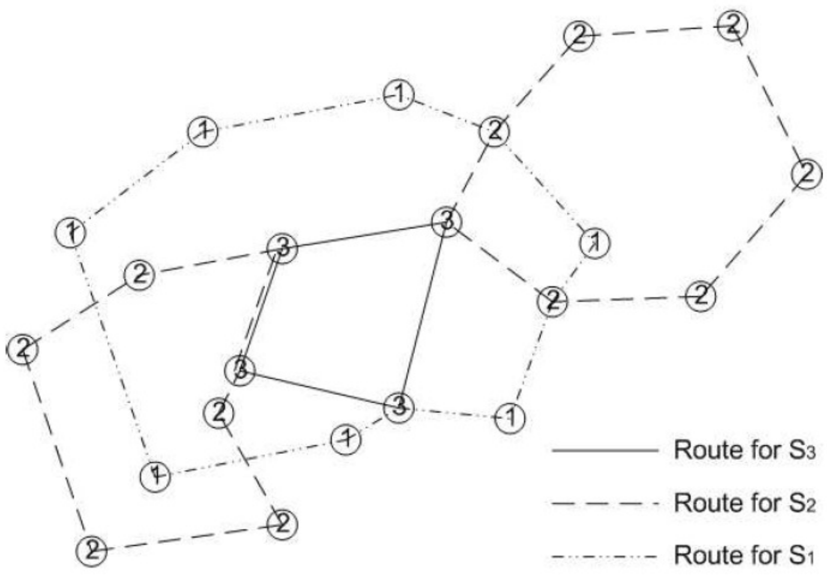
- The M spanning trees result in M groups of nodes to be walked through by distinct actuators
- The route design problem can be reduced to traveling salesman problem (TSP) for each group of nodes
- We adopt the Approx-TSP-Tour algorithm here, which use MST to create a tour and perform a preorder traversal on the tree to obtain a Hamiltonian cycle



T1 T2 T3

Route Design Algorithm by Varying Path Length (RDPL)

- Sensors with similar weights will be assigned to the same route with a particular length
- Sensors with higher W_i will be put on shorter routes, and vice versa



(1) Clustering MSTs with Different W_i

- We divide the weight $W_i \in [0, 1]$ into *w* ranges
- Sensors are grouped into the appropriate sets of sensors, $S_1, S_2, \ldots, S_w$, according to their weight ranges
- The number of actuators assigned to the corresponding set of sensors are $a_1, a_2, \ldots, a_w$
- The ideal inter-arrival time for sensors $S_1, S_2, \ldots, S_w$, could be $w*T, (w-1)*T, \ldots, 2*T, T$
- The cost $C(T_k)$ represents the length of the route
- $C(T_k)/a_k$ is proportional to the inter-arrival time $(w+1-k)*T$
- A minimum spanning tree T_k is then constructed for each set of sensors and its corresponding cost $C(T_k)$ is calculated.
- The cost $C(T_k)$ is a good approximation to the length of route *k*

(2) Estimate Number of Actuators for each T_k

$$\begin{aligned} C(T_1)/a_1 &= w * T * v \\ C(T_2)/a_2 &= (w-1) * T * v \\ &\dots \\ C(T_w)/a_w &= T * v, \end{aligned}$$

and

$$a_1 + a_2 + \dots + a_w = M$$

From the above equations, we achieve

$$\frac{C(T_1)}{w * T * v} + \frac{C(T_2)}{(w-1) * T * v} + \dots + \frac{C(T_w)}{T * v} = M,$$

hence,

$$T * v = \{\sum_{k=1}^{w} C(T_k)/(w+1-k)\}/M$$

and

$$\begin{aligned} a_k &= C(T_k)/(T * v * (w+1-k)) \\ &= \frac{C(T_k) * M}{(w+1-k)\{\sum_{k=1}^{w} C(T_k)/(w+1-k)\}} \end{aligned}$$

(3) Allocate the Actuators

- The a_k achieved above is an ideal value, so we need to determine the practical number of actuators n_k
- If a_k is smaller than 1, the sensors in S_k will be accumulated in $accum_S$ with other sensors in the following weight ranges, until $accum_S$ is greater than 1.
- A new route R_q is then formed with n_q = round($accum_S$), including all the sensors in $accum_S$
- Similar to RDNV, the route design problem can then be reduced to the Travelling Salesman Problem (TSP) for each route.

- Distributed implementation on RDPL (D-RDPL)

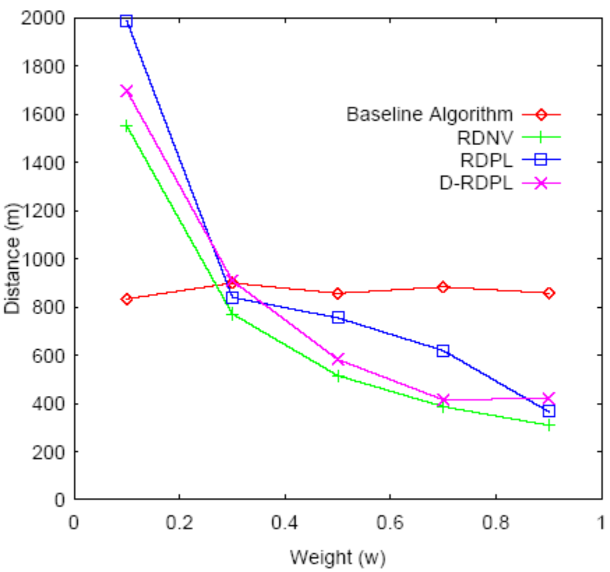
Performance Evaluation

- Algorithms:
  - Baseline Algorithm: provides one actuator to each route for a sensor group with the same weight
  - RDNV
  - RDPL
  - D-RDPL

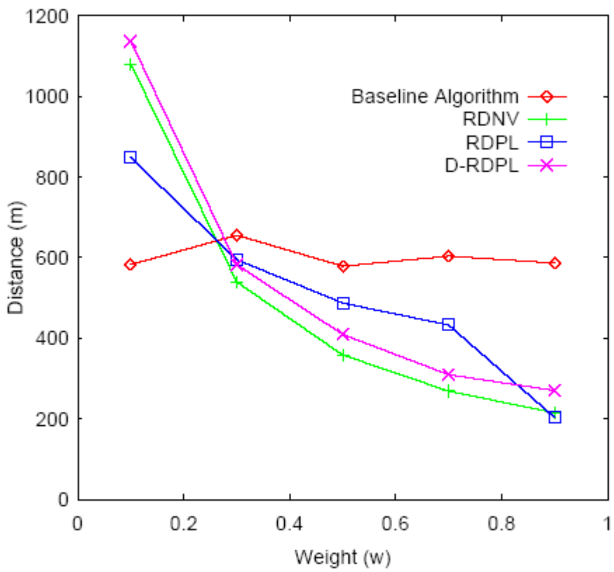
Network size	200m x 200m
Sensor distribution	Uniform random or Cluster-based uniform or Cluster-based non-uniform
No. of sensors (N)	100
Weight of sensors (W_j)	0.0-1.0
No. of actuators (M)	5 or 8
Speed of actuators	v
Radio range	40m
MAC layer	IEEE 802.11

Average Inter-arrival Distance

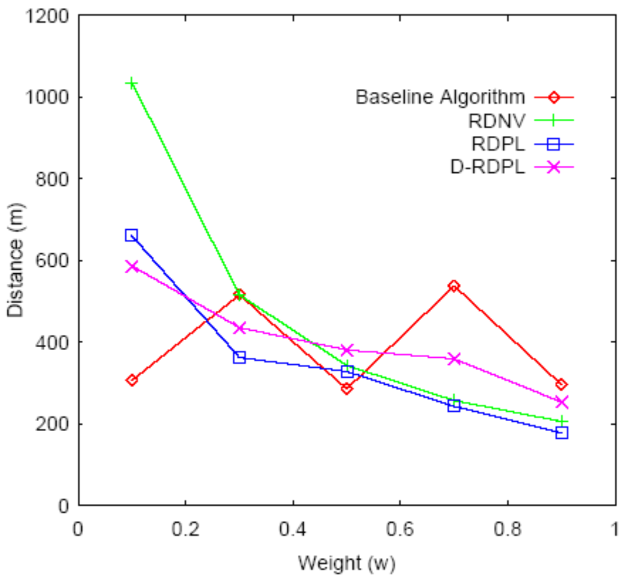
N = 100, M =5



Uniform random



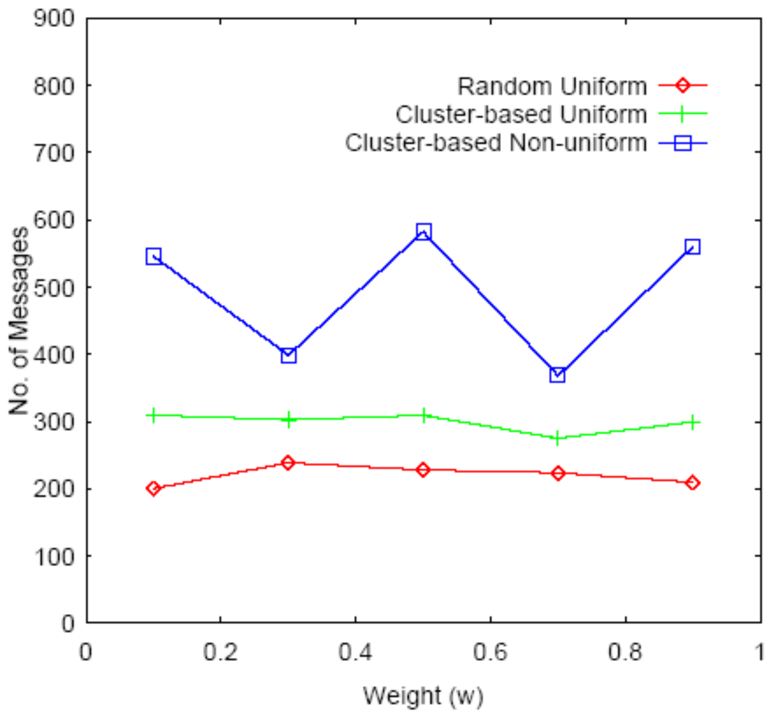
Cluster-based Uniform



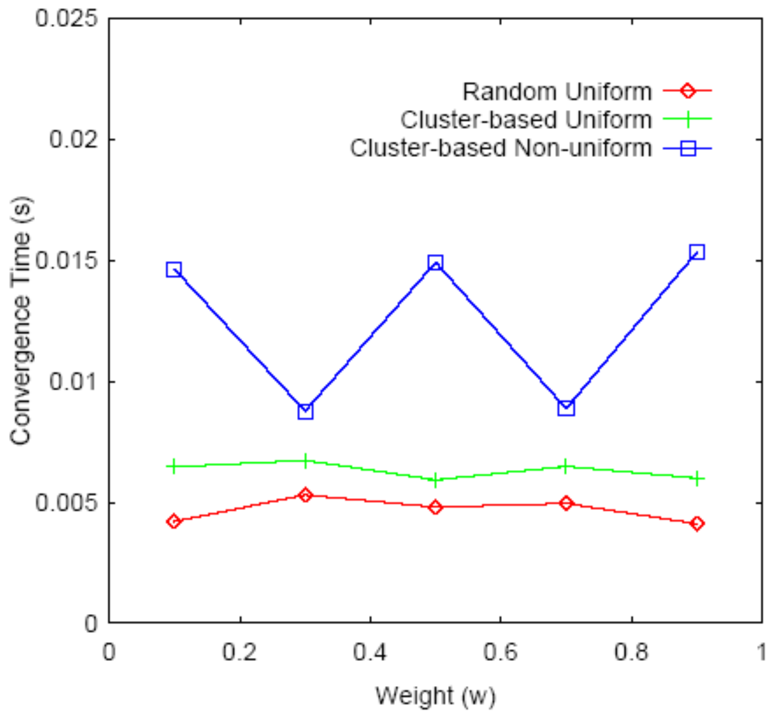
Cluster-based non-uniform

D-RDPL

N = 100, M =5



Message Overhead

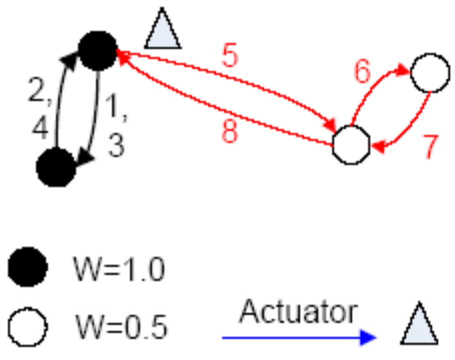


Convergence Time

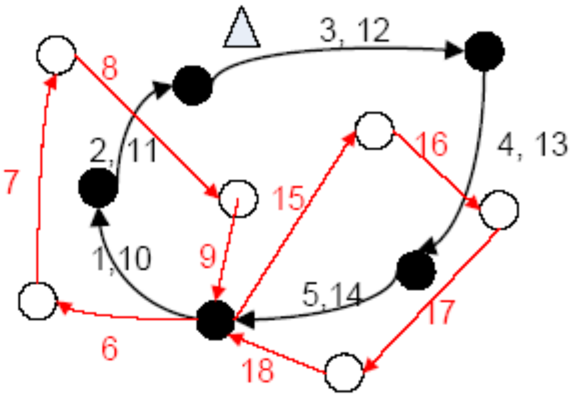
Adaptive Delay-Minimized Route Design

- To minimize the data collection time in a stochastic and dynamically changing sensing environment
- We propose a probabilistic route design algorithm (PROUD) for wireless sensor-actuator networks
- This is a departure from the previous static and deterministic mobile element scheduling problems
- PROUD offers delay-minimized routes for actuators and adapts well to network dynamics and sensors with non-uniform weights
- This is achieved through a probabilistic visiting scheme along pre-calculated routes

Example on RDP



(a)



(b)

$$Minimize \sum_{\forall i} A_i * w_i * N_i$$

$$A_w = \frac{2|TSP(S_b)| + |TSP(S_w)| + 2 * ||S_w, S_b||}{v}$$

Small-scale Network

- (1) Form a Priori Route
  - A priori route is formed by constructing a TSP path which contains all locations to be visited
  - We adopt the well-known Approx-TSP-Tour algorithm here for its low cost and bounded performance
- (2) Visit sensors probabilistically
  - Actuators visit the sensors on the priori route in sequence probabilistically
  - We set the visiting probability p_i of a location i to be w_i, where w_i is the (normalized) weight of the sensors
  - Sensors with higher weights should be assigned with a higher probability

Small-scale Network

- (3) Allocate the actuators
  - The expected route length with probabilistic visiting can be calculated as

$$E[R] = \sum_{r=0}^{n-2} \sum_{i=1}^{n} \|i, i+r\| * p_i * p_{i+r+1} \Pi_{k=1}^{r} (1 - p_{i+k})$$

  - For a sensor *i* with a visiting probability p_i, its average actuator inter-arrival time A_i is thus

$$A_i = \frac{E[R]}{p_i * v * M}$$

Analysis

- Time complexity analysis:

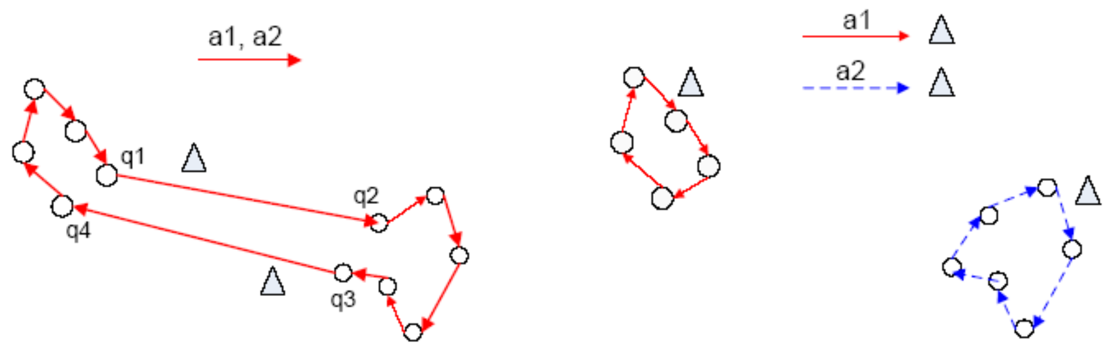
$$O(N^2 + M)$$

- Bound analysis:

$$\frac{A_i}{A_i^*} \leq \frac{E[R]}{p_i * |TSP(S_i)|}$$

Large-scale Network

- (1) Form clusters
  - It divides an MST into two sub-trees by removing its longest edge *e*
- (2) Form priori routes and visit sensors probabilistically
- (3) Allocate actuators
  - Routes with longer expected lengths should be allocated with more actuators



(a) (b)

Load Balancing

- Multi-route improvement algorithm
  - Consider two routes R_1 and R_2 involved in multi-route improvement.
  - Their new expected route lengths become ideal if $E[R'_1] = E[R'_2] = (E[R_1] + E[R_2])/2$.
  - We provide an approximation method to transfer a proportion of sensor locations from MST_1 to MST_2

$$\frac{cost(\xi)}{cost(MST_1)} = \frac{(E[R_1] - E[R_2])/2}{E[R_1]}$$

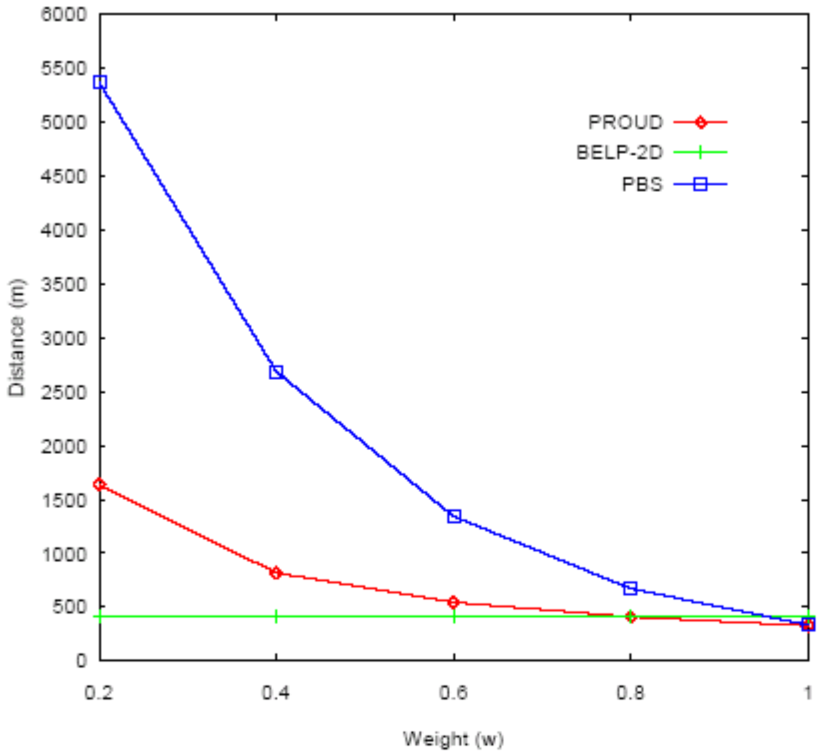
- Task exchange algorithm

Algorithm 13 Task exchanges among actuators

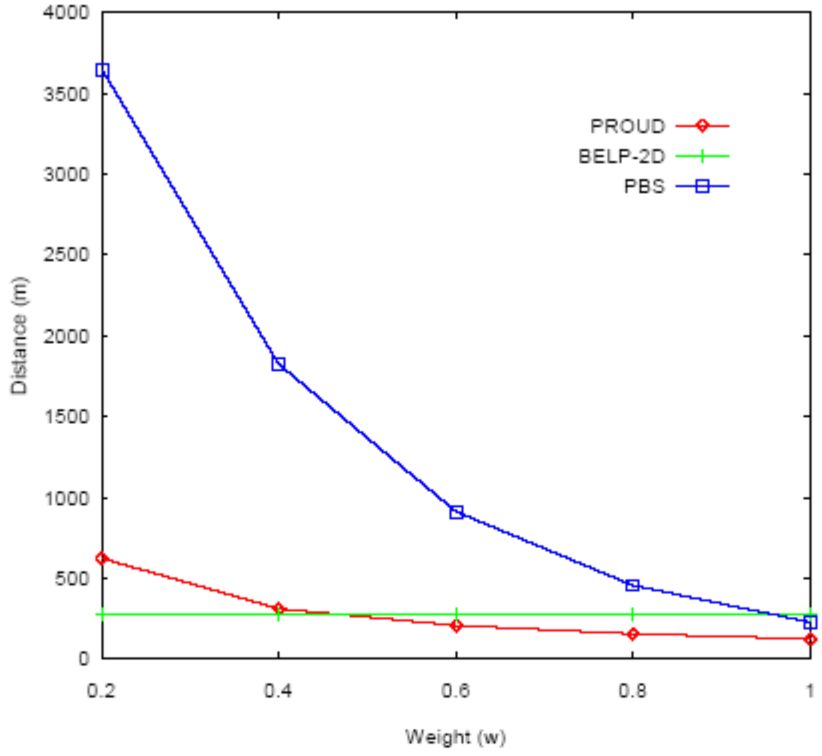
```
if (Energy_A1 << Energy_A2)and(v1 >> v2) then
    A1 moves to R2;
    A2 moves to R1;
end if
```

Actuator Inter-arrival Time

N = 100, M =5

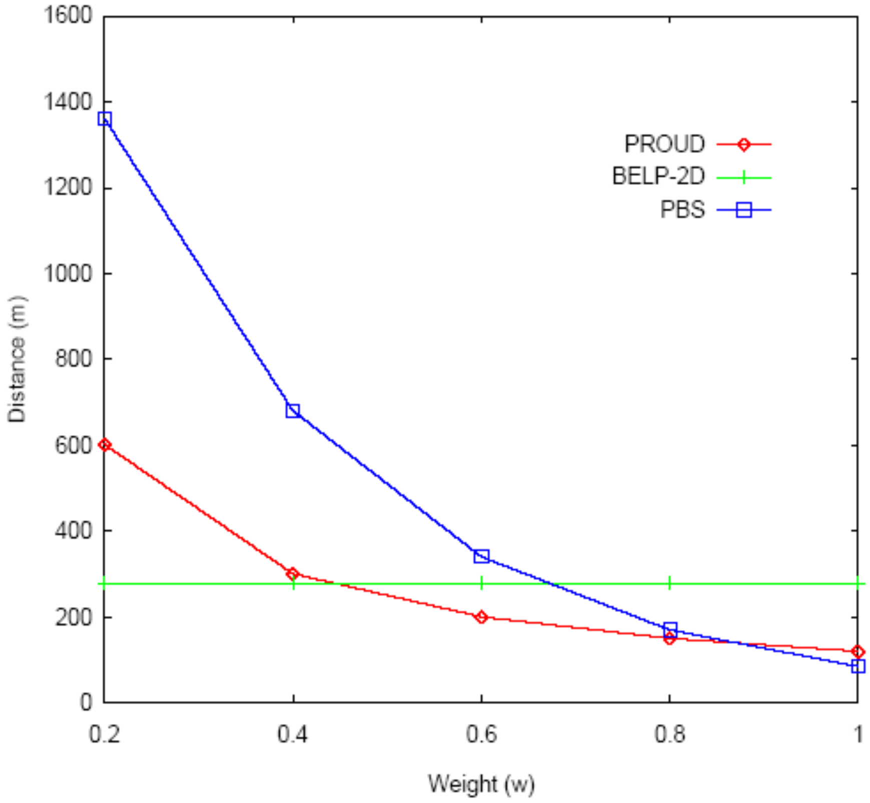


Uniform random

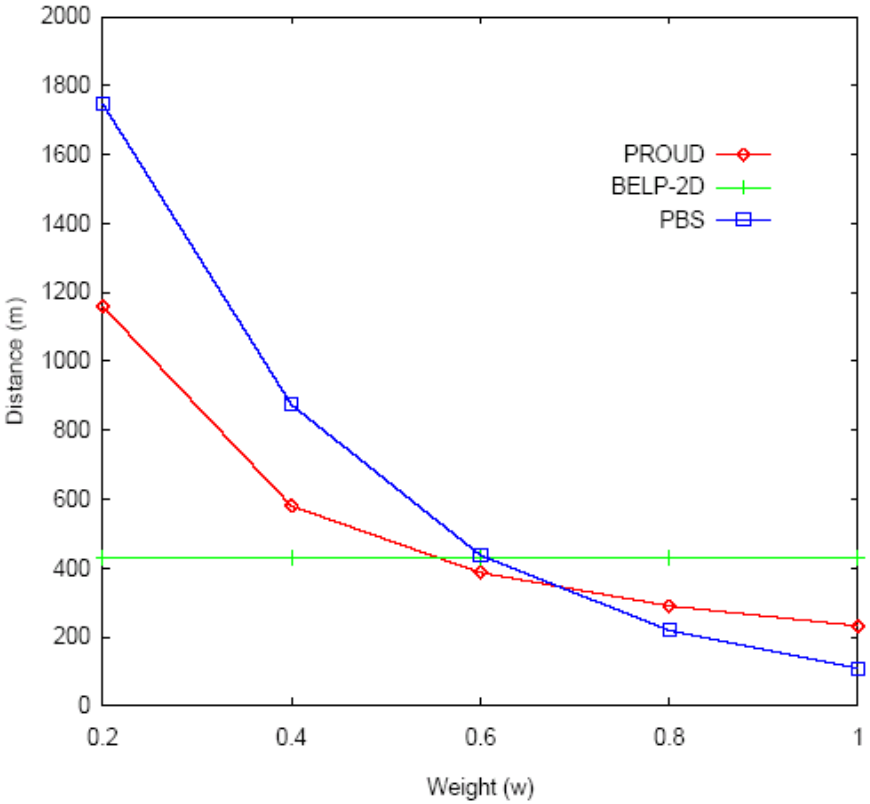


Cluster-based uniform

Actuator Inter-arrival Time



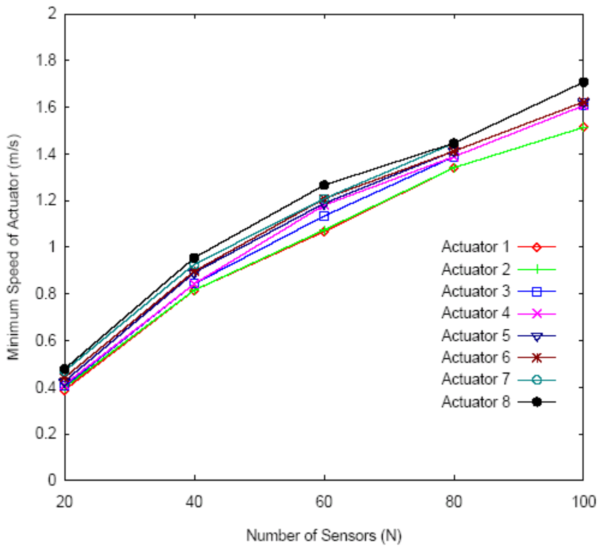
Cluster-based non-uniform



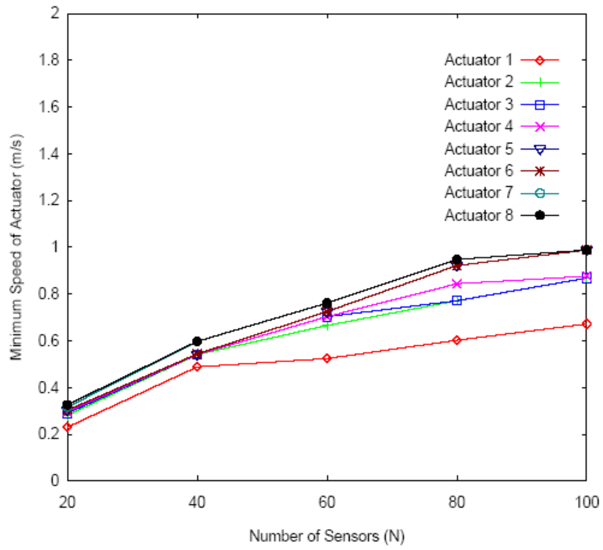
Eye Topology

Minimum Actuator Speed

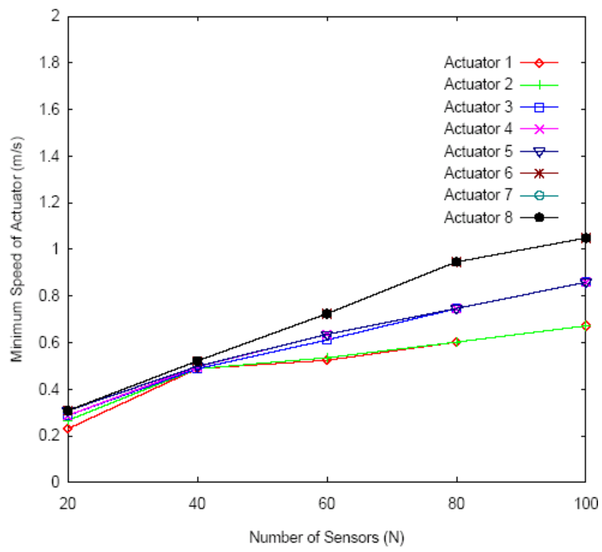
N = 100, M =5



Uniform random

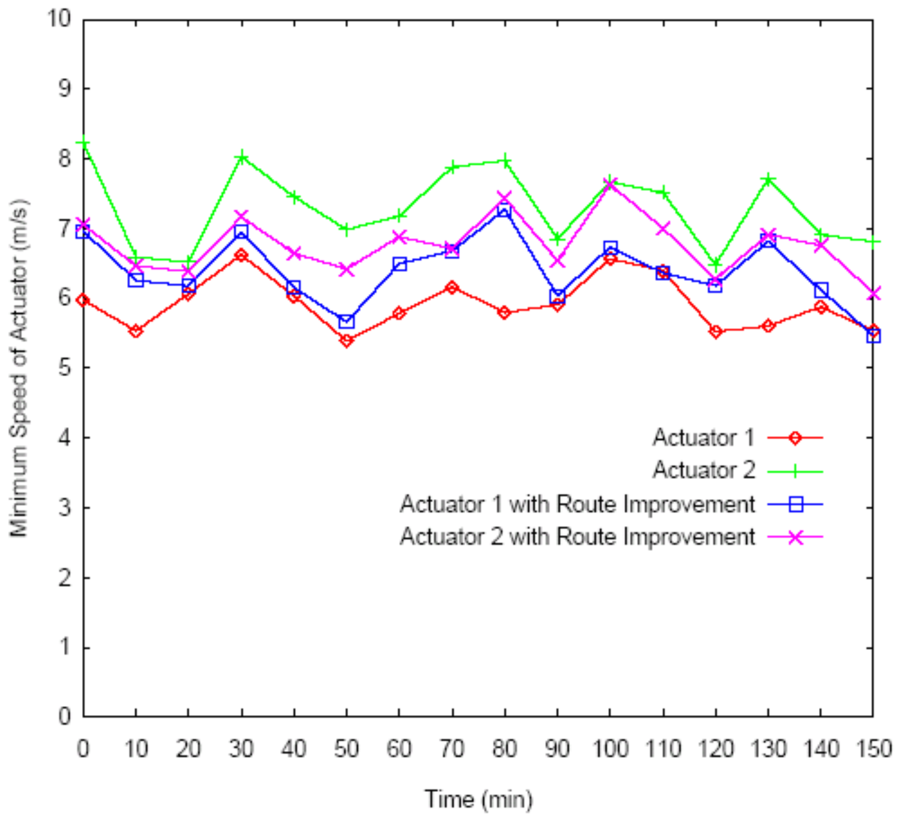


Cluster-based uniform

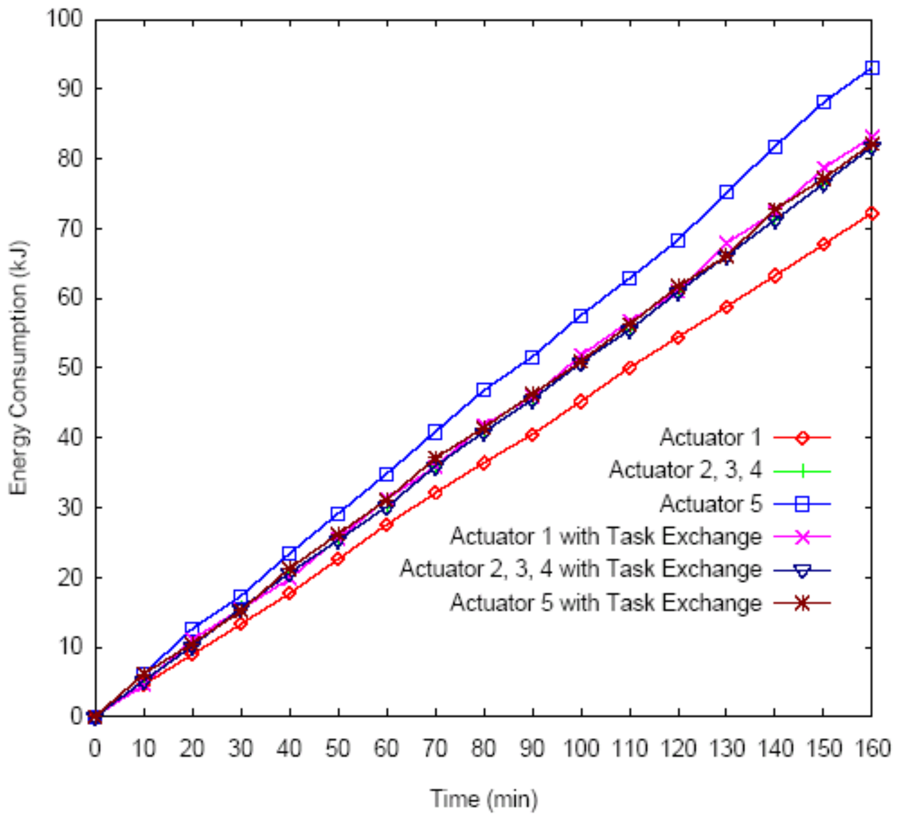


Cluster-based non-uniform

Load Balancing



Multi-route improvement



Task exchange

Summary of Part 3

- We focused on WSN with multiple actuators and their route design
- We demonstrated the problem is NP-hard and proposed an effective MST-based approximation algorithm
- It aims at minimizing the overall inter-arrival time of the actuators
- It differentiates the visiting frequency to sensor locations with different weights
- We further proposed an adaptive route design algorithm (PROUND)
- Simulation results suggested that the algorithm remarkably reduces the average inter-arrival time

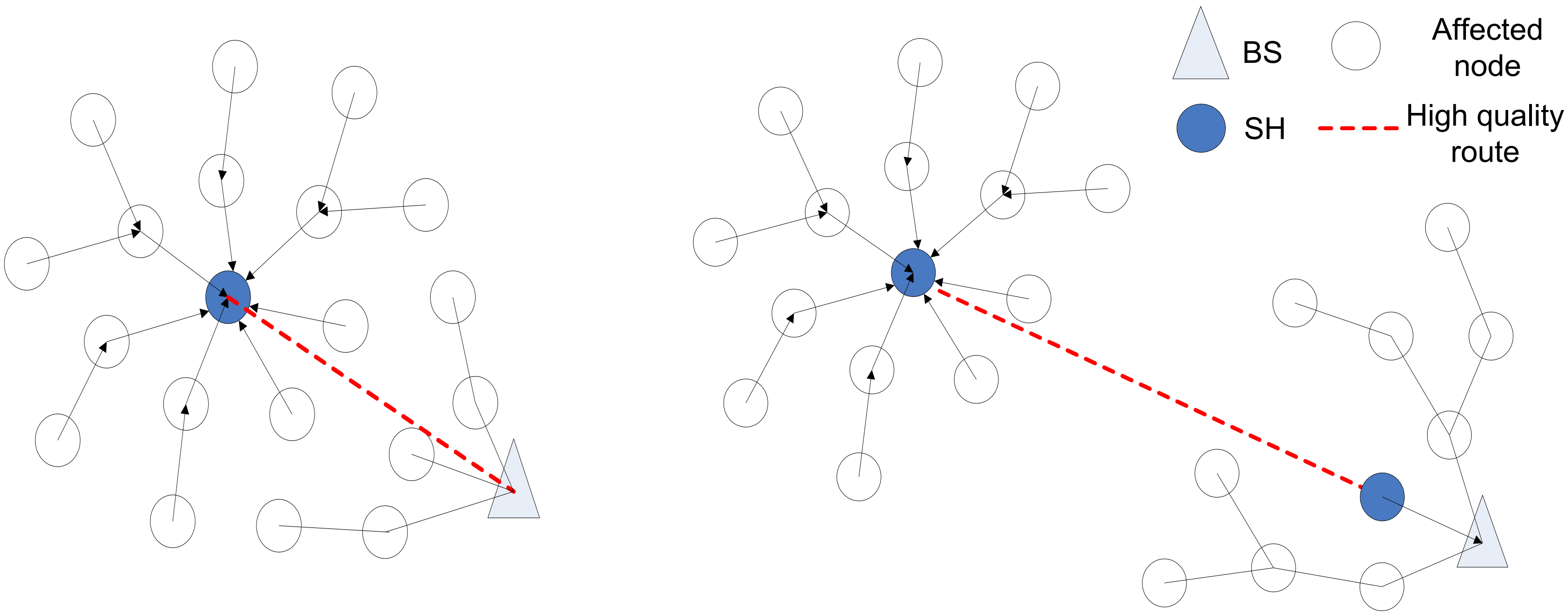
Intruder Detection Against Sinkhole Attack in Wireless Sensor Networks

Part 4

Sinkhole Attack

- Many-to-one communication
  - Vulnerable to the *sinkhole attack*
- Prevent the base station from obtaining complete and correct sensing data
- Particularly severe for wireless sensor networks
- Some secure or geographic based routing protocols resist to the sinkhole attacks in certain level
- Many current routing protocols in sensor networks are susceptible to the sinkhole attack

Sinkhole Attack



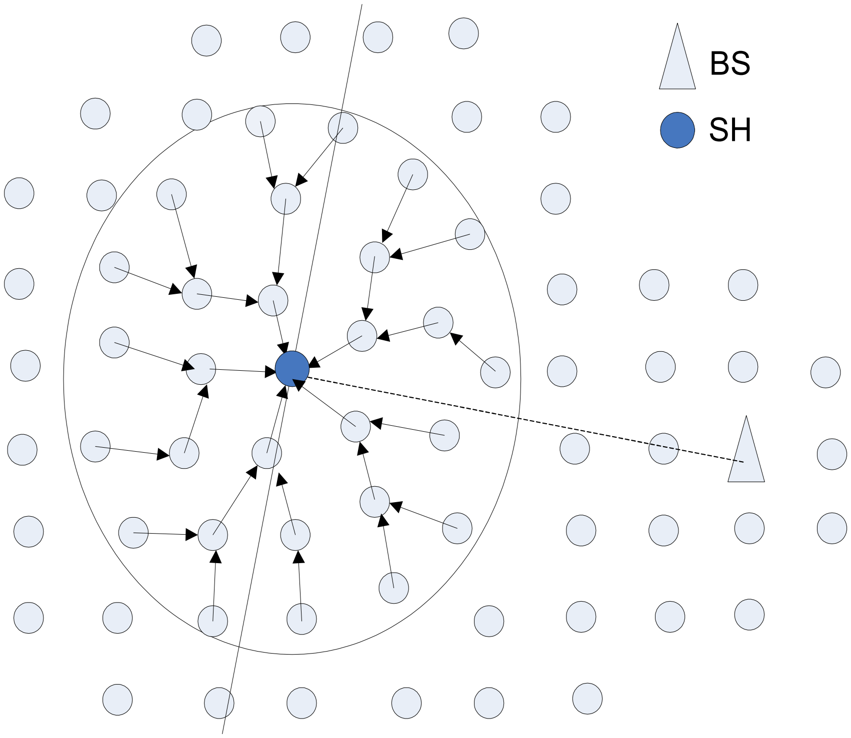
- Left: using an artificial high quality route
- Right: using a wormhole

Our Work

- Propose an algorithm for detecting sinkhole attacks and identifying the intruder in an attack
  - Base station collects the network flow information with a distributed fashion in the attack area
  - An efficient identification algorithm that analyzes the collected network flow information and locate the intruder
- Consider the scenario that a set of colluding nodes cheat the base station about the location of the intruder

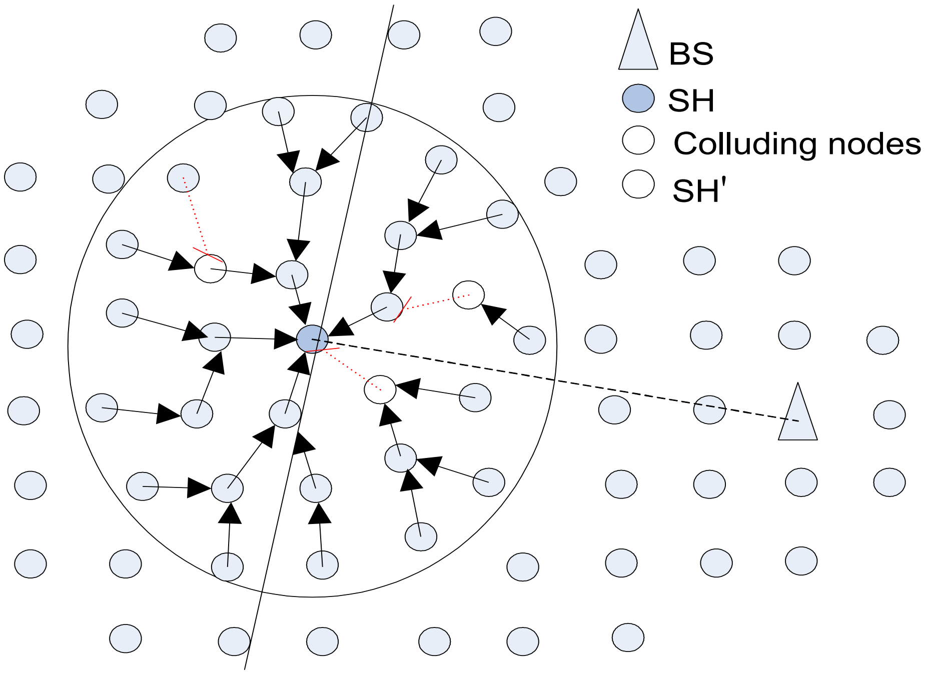
Identifying the Intruder

- Network flow information can be represented by a directed edge
- Realizes the routing pattern by constructing a tree using the next hop information collected
- An invaded area possesses special routing pattern
  - All network traffic flows toward the same destination, which is compromised by the intruder *SH*

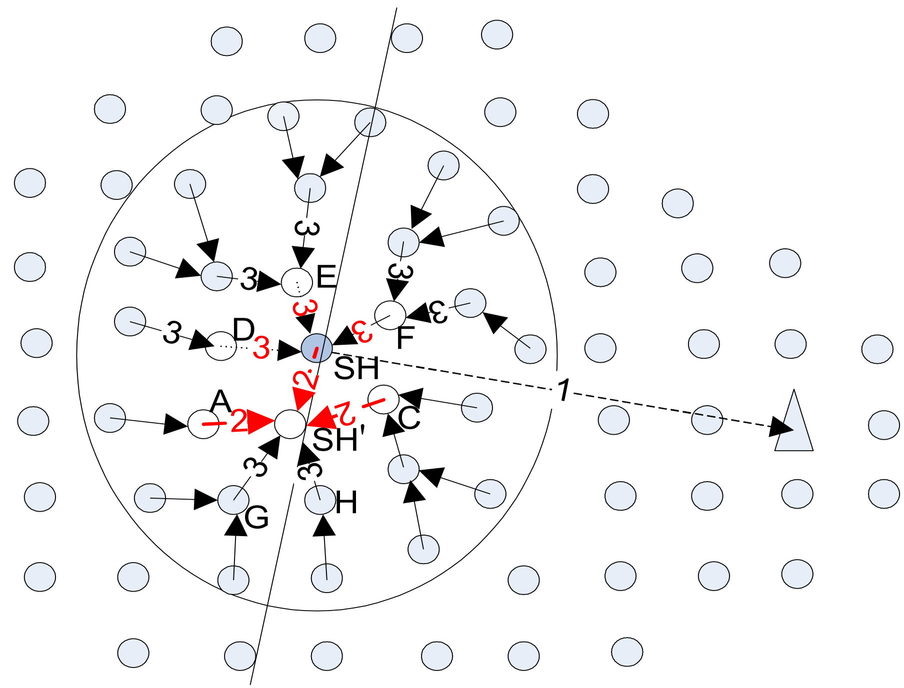


Multiple Malicious Nodes

- Drop some of the reply packets



- Provide incorrect flow information



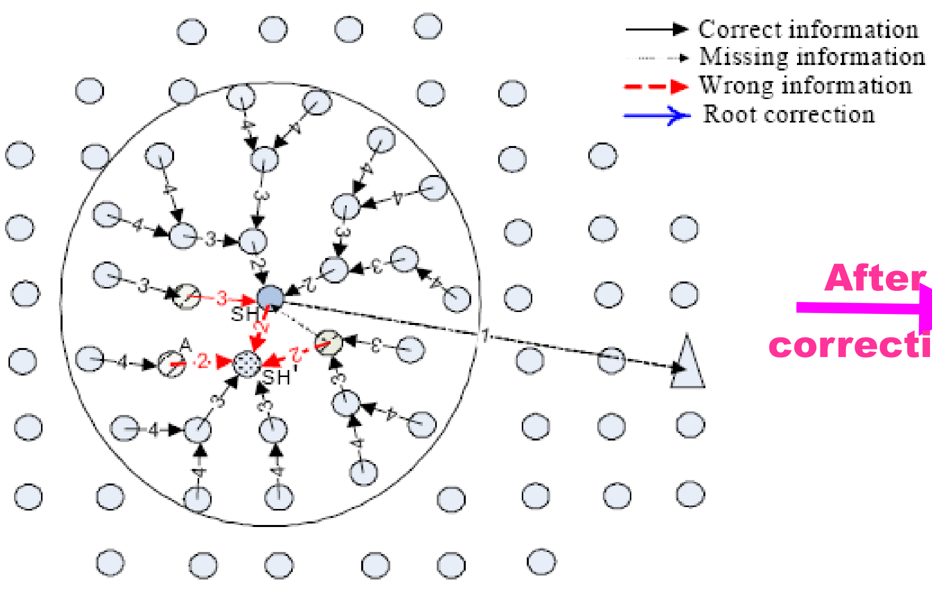
Their objective is to hide the real intruder *SH* and blame on a victim node *SH’*

Dealing with Malicious Nodes

- Maintain an array *Count*[]
  - Entry *Count*[*i*] stores the total number of nodes having hop count difference *i*
  - Index *i* can be negative (The node distance is smaller than its actual distance from the current root)
- If *Count*[0] is not the dominated one in the array, it means the current root is unlikely the real intruder

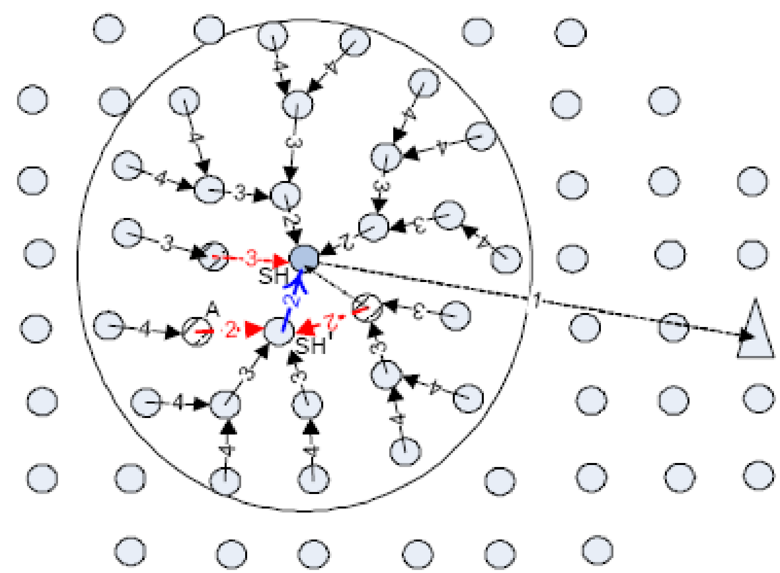
Root Correction Example

- Eventually, node SH becomes the new root:



i	-2	-1	0	1	2
Count[i]	0	14	8	6	0

After correction



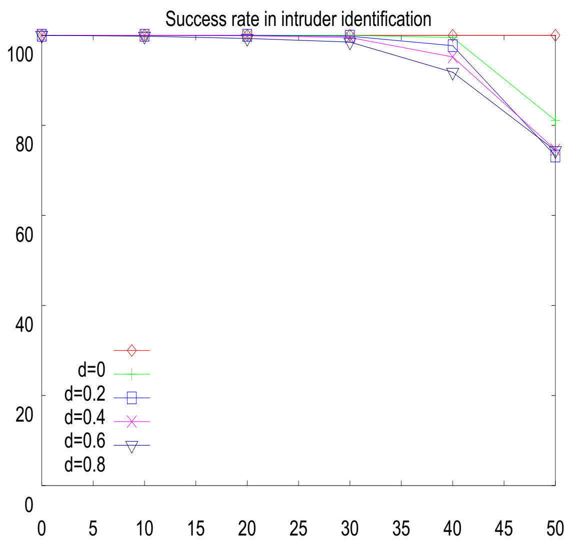
i	-2	-1	0	1	2
Count[i]	0	1	21	6	0

Performance Evaluation

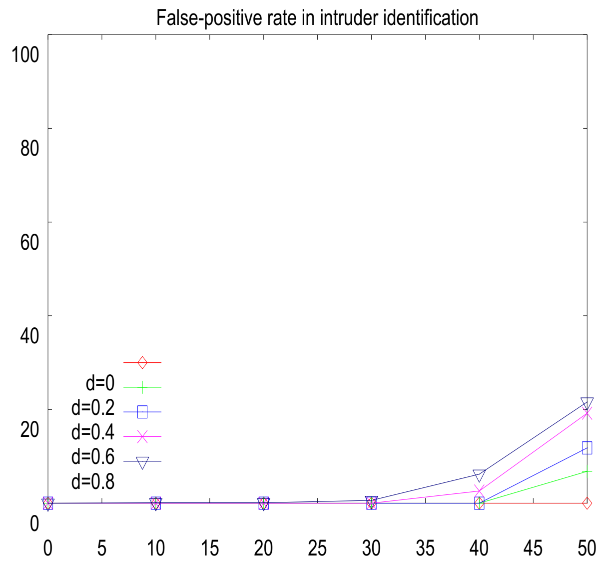
- Accuracy of Intruder Identification
  - Success Rate
  - False-positive Rate
  - False-negative Rate
- Communication Cost
- Energy Consumption

No. of nodes in network	400
Size of network	200m x 200m
Transmission range	10m
Location of *BS*	(100,100)
Location of sinkhole	(50, 50)
Percentage of colluding codes (*m*)	0 – 50%
Message drop rate (*d*)	0 – 80%
No. of neighbors which a message is forwarded to (*k*)	1 – 2
Packet size	100bytes
Max. number of reply messages per packet	5

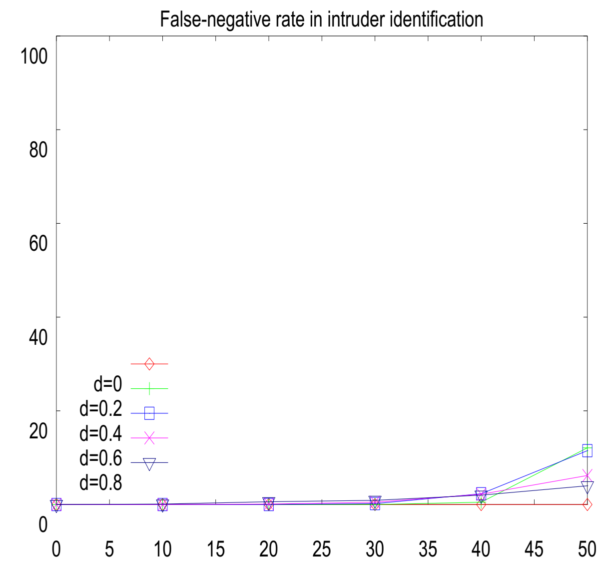
Intruder Identification



Success rate

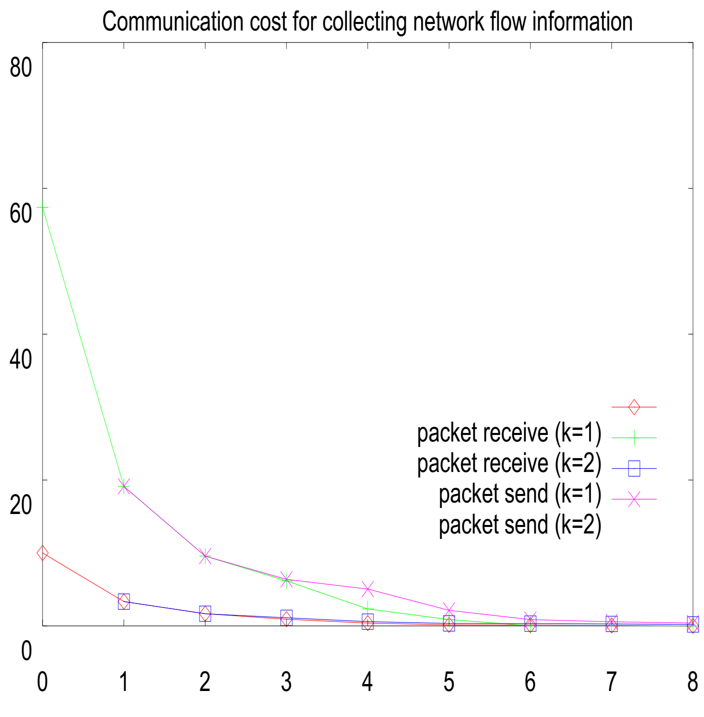


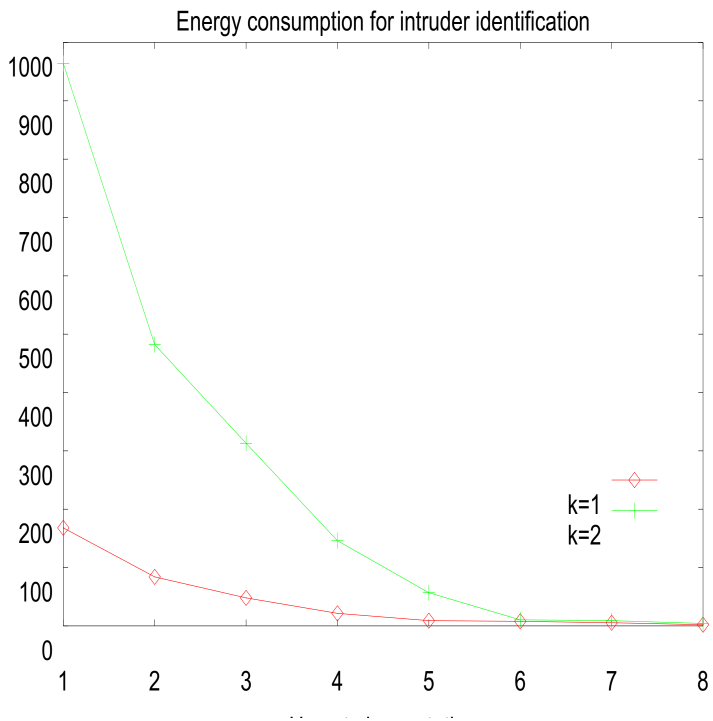
False-positive rate



False-negative rate

Communication Cost and Energy Consumption





Summary of Part 4

- We proposed an effective method for identifying sinkhole attack in wireless sensor networks
- It locates a list of suspected nodes by checking data consistency, and then identifies the intruder in the list through analyzing the network flow information
- A series of enhancements is proposed to deal with cooperative malicious nodes which attempt to hide the real intruder
- Numerical analysis and simulation results are provided to demonstrate the effectiveness and accuracy of the algorithm
- We are interested in more effective statistical algorithms for identifying data inconsistency

Conclusions

- We investigated real-time communication and coordination in WSAN
- We proposed a general delay-aware reliability-centric framework for event reporting in WSAN
- We provided fault-tolerant and energy efficient data transport protocol for real-time data collection
- We studied the route design problem and proposed delay-minimized and cooperative solution to coordinate actuators and collect data efficiently
- We investigated the security issues in sensor networks and proposed an effective algorithm for intruder identification against Sinkhole attack

Selected Publications

Journal Paper

- **Edith C.-H. Ngai**, Jiangchuan Liu, and Michael R. Lyu, "An Efficient Intruder Detection Algorithm Against Sinkhole Attacks in Wireless Sensor Networks," to appear in *Computer Communications*, 2007.
- **Edith C.H. Ngai** and Michael R. Lyu, "Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks," to appear in*International Journal of Wireless and Mobile Computing*.

Conference Paper

- **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," *IEEE Global Telecommunications Conference (Globecom'07),* , Washington, DC, USA, November 26-30 , 2007.
- **Edith C.-H. Ngai**, Jiangchuan Liu, and Michael R. Lyu, "An Adaptive Delay-Minimized Route Design for Wireless Sensor-Actuator Networks," *The 4th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'07),* , Pisa, Italy, October 8-11, 2007.
- **Edith C.H. Ngai**, Jiangchuan Liu, and Michael R. Lyu, "Delay-Minimized Route Design for Wireless Sensor-Actuator Networks," *IEEE Wireless Communications & Networking Conference (WCNC'07)*, Hong Kong, March 11-15, 2007.
- 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," *IEEE Wireless Communications & Networking Conference (WCNC'07)*, Hong Kong, March 11-15, 2007.
- **Edith C.H. Ngai**, Yangfan Zhou, Michael R. Lyu, and Jiangchuan Liu, "Reliable Reporting of Delay-Sensitive Events in Wireless Sensor-Actuator Networks," *The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'06)*, Vancouver, Canada, October 9-12, 2006.
- 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," *International Wireless Communications and Mobile Computing Conference (IWCMC'06)*, Vancouver, Canada, July 3 - 6, 2006.
- **Edith C.H. Ngai**, Jiangchuan Liu, and Michael R. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks," *IEEE International Conference on Communications (ICC'06)*, Istanbul, Turkey, June 11-15, 2006.
- **Edith C.H. Ngai** and Michael R. Lyu, "An Authentication Service Based on Trust and Clustering in Wireless Ad Hoc Networks: Description and Security Evaluation," *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, Taichung, Taiwan, June 5-7, 2006.
- **Edith C.H. Ngai**, Michael R. Lyu, and Jiangchuan Liu, "A Real-Time Communication Framework for Wireless Sensor-Actuator Networks," *IEEE Aerospace Conference 2006*, Big Sky, Montana, USA, March 4-11, 2006.

Future Directions

- More advanced actuator communication schemes can be explored
- Communication/storage points can be investigated to enhance data collection
- Different hardware facilities and MAC layer protocols of sensors can be studied and experimented
- More security mechanisms can be explored to protect the network

Q & A

Thank You!