GPS-free Disaster-Scale Mapping and Energy-Efficient Alerting Scheme in a Wireless Sensor Network

Ehssan Sakhaee, Naoki Wakamiya, Masayuki Murata
Graduate School of Information Science and Technology
Osaka University, Japan

The Second International Conference on Sensor Technologies and Applications
SENSORCOMM 2008, August 25-31, Cap Esterel, France.

Overview

- Aims and assumptions
- Basic approach of disaster reporting
- HexNet mapping scheme
- Proposed protocol for disaster reporting
- Features of the protocol
- HexNet routing algorithm
- Simulations
- Conclusion

Motivation

- Monitoring of remote disaster-prone environments using sensor networks.
- Reporting the disaster's scale (size of disaster) and location.
- Sensor nodes are destroyed!
  - Must be kept cheap and easily replaceable.

Aims and Assumptions

- A sensor network which reports disaster-scale (size) and location
- Sensor nodes are not equipped with global positioning system (GPS)
  - reduce cost of the sensors as they are prone to destruction in the targeted application scenario.
- A *mapping scheme* for sensors in order for the sink to effectively determine location and scale of disaster.
- Sensor nodes are static and their placement follows rules, outlined in the *mapping scheme.*
- Sensor nodes perform one-hop transmission.

Basic Approach

- When a disaster occurs, the nodes in the affected region are destroyed.
- The nodes surrounding the affected area notice the disaster by death of their neighbor nodes that resided in the affected area.
- Nodes which notice the disaster issue *alert* messages which are forwarded to the sink via other intermediate nodes.
- Only nodes which physically sense the disaster and neighbor nodes leading in the direction of the sink take part in routing alerts.

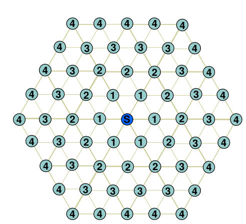
Mapping Scheme

• Sensors are arranged at equal distances from each other in a hexagonal-style grid.

• Sensors closer to the sink have a lower Range ID than nodes further away.

• The sensors' Range IDs increment with increasing distance from sink.

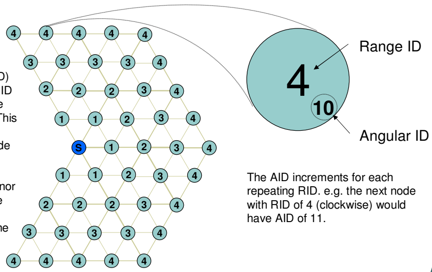
• The position of individual sensors is determined by a combination of two identifiers involving a Rang ID and Angular ID.



Range ID and Angular ID

The node's Range ID (RID) together with the Angular ID (AID) is used to determine the location of the node. This combination gives the precise location of the node relative to the sink.

Although neither the RID nor AID is unique by itself, the combination of the two is unique for each node in the network.



The AID increments for each repeating RID. e.g. the next node with RID of 4 (clockwise) would have AID of 11.

The Mapping System and Angular ID Calculation

AID = 3n(1 - θ/180) + 1

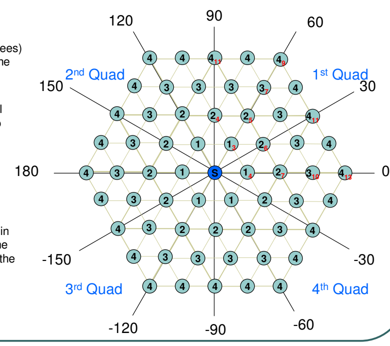
where θ is the angle (in degrees) relative to the sink and n is the RID of the node.

Using the sine rule, the actual physical distance from sink to node x is

$$D = \frac{nd\sqrt{3}}{2\sin(\theta+60)}$$

where n is node x's RID, d is distance between each node in the network (constant), θ is the relative angle of node x from the sink given by

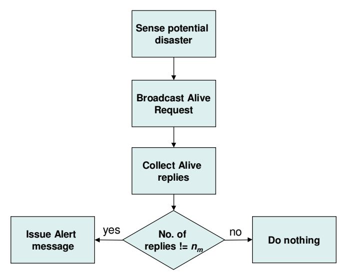
$$\theta = \frac{60}{n}(3n - \theta_{ID} + 1)$$



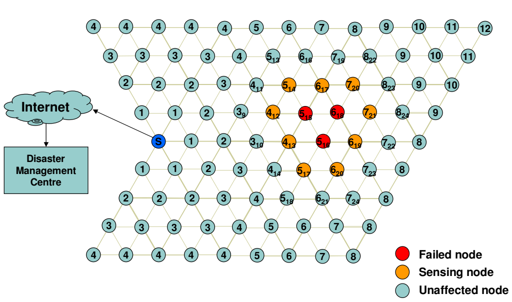
Disaster Reporting Scheme Overview

- When a sensor node "senses" a disaster, it will respond to this by broadcasting an "alive request" to its neighboring nodes. Neighboring (1-hop) nodes that receive this message would then reply to this alive request stating their "alive" status.
- Sensors *know* the number of neighbors in their vicinity n_m (6 in the normal case) and hence the number of replies they are suppose to receive upon their alive request.
- If a sensor node does not receive n_m replies, where n_m is the number of neighbors, it will issue a Alert (disaster) message to its neighbors containing the node id of nodes which have not replied. Neighbors which are closer to the sink than the node issuing the disaster message will rebroadcast the message unless *all* better nodes are destroyed, in which case the next best node either forwards or decides to simply not forward the message*. This continues until the sink receives the disaster messages (from nodes surrounding the disaster area).
- The sink then sends all the messages to the appropriate authority centre (via the Internet/satellite etc.) where the region of disaster and disaster scale is determined using knowledge of all nodes which have been destroyed.

Disaster Reporting Scheme Flowchart



Reporting of Disaster



Sensing and Reporting

*Neighbor node's responses to the Alive-request messages.

Sensing Node	Expected Replies	Actual Replies
5_{14}	$4_{11}5_{13}6_{16}6_{17}5_{15}4_{12}$ (6)	$4_{11}5_{13}6_{16}6_{17}4_{12}$ (5)
6_{17}	$5_{14}6_{16}7_{19}7_{20}6_{18}5_{15}$ (6)	$5_{14}6_{16}7_{19}7_{20}$ (4)
7_{20}	$6_{17}7_{19}8_{22}8_{23}7_{21}6_{18}$ (6)	$6_{17}7_{19}8_{22}8_{23}7_{21}$ (5)
7_{21}	$6_{18}7_{20}8_{23}8_{24}7_{22}6_{19}$ (6)	$7_{20}8_{23}8_{24}7_{22}6_{19}$ (5)
6_{19}	$5_{16}6_{18}7_{21}7_{22}7_{23}6_{20}$ (6)	$6_{18}7_{21}7_{22}7_{23}6_{20}$ (5)
6_{20}	$5_{17}5_{16}6_{19}7_{23}7_{24}6_{21}$ (6)	$5_{17}6_{19}7_{23}7_{24}6_{21}$ (5)
5_{17}	$4_{14}4_{13}5_{16}6_{20}6_{21}5_{18}$ (6)	$4_{14}4_{13}6_{20}6_{21}5_{18}$ (5)
4_{13}	$3_{10}4_{12}5_{15}5_{16}5_{17}4_{14}$ (6)	$3_{10}4_{12}5_{17}4_{14}$ (4)
4_{12}	$3_{9}4_{11}5_{14}5_{15}4_{13}3_{10}$ (6)	$3_{9}4_{11}5_{14}4_{13}3_{10}$ (5)

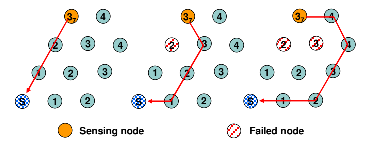
Node Types and their function

Node Type	Function
Failed/Destroyed Node	Nothing.
Sensing Node	1. Broadcast *alive request* messages 2. Collect "alive" messages 3. Send Alert to neighbors
Forwarding Node	1. Check the Alert message to see whether it is the best* node to forward. If so, broadcast, otherwise ignore.
Sink	Forward messages to Disaster Management centre.

Hop Prediction to Sink

- A node can *predict* the number of hops to the sink from the knowledge of its failed neighbor nodes.
- The prediction is manifested and updated in an integer field called the *Hop Count Predictor* (HCP) in the Alert message header. The HCP is defined as follows.

1. HCP is set to the RID initially
2. It is incremented by 1 if the next best node for forwarding has a RID equal to the parent node.
3. It is incremented by 3 if the next best node for forwarding has a RID *greater* than the parent node.



Cost Metric

The cost of node *X* to the sink is

$$Cost_x = w_1 E_x^{-1} + w_2 HCP_x + w_3 D_x$$

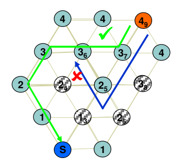
where E_X is the current *normalized* energy of node *X*, HCP_X is the HCP of node *X* to the sink, D_x is the physical distance of node *X* to the sink (calculated using RID_{AID}), w_1 and w_2 are the weights for the *significance* of (individual node) energy and delay in routing respectively, and $0 \le w_1, w_2 \le 1$.

Local Neighbor Node Table (LNNT)

Each node posses a local neighbour node table (LNNT) containing information about its neighbour, such as their status, HCP, and the neighbours which are within a 1-hop range of each other.

LNNT at node 2_5

Node	Neighbors	Status	HCP	Energy
2_5	$1_3 2_4 3_6 3_7 3_8 2_6$	ALIVE	5	10
1_3	$2_4 2_5$	FAIL	N/A	N/A
2_4	$3_6 1_3$	FAIL	N/A	N/A
3_6	$2_4 3_7$	ALIVE	4	10
3_7	$3_6 3_8$	ALIVE	5	10
3_8	$3_7 2_6$	FAIL	N/A	N/A
2_6	$1_3 3_7$	FAIL	N/A	N/A



In this case, 2_5 has a HCP of 5. It *knows* that one of its neighbors namely 3_6 has a lower HCP of 4. When it receives an Alert from 4_9 via 3_7 it WILL NOT BROADCAST it as it knows that 3_6 (being a neighbor of 3_7) can broadcast it with a smaller cost as it has a smaller HCP.

HexNet Routing Algorithm

- 1) A sensor node *X* senses a physical change in its environment (heat/sound/electrical surge), which could be caused by a potential disaster.
- 2) Node *X* broadcasts an Alive Request (AREQ) to its 1-hop neighbors, and initiates a timer for Alive Reply (AREP) collection.
- 3) If node *X* also receives an AREQ, it will wait until it receives AREPs from its neighbors.
- 4) Once all the expected number of AREPs are received from *X*'s *alive* neighbors, or when the timeout is reached, node *X* will calculate its own HCP using knowledge of failed neighbors, then append this value to the AREP and broadcast it. At this time, the LNNT is updated.
- 5) Node *X* will then use the collected information (parameters shown in LNNT) to calculate the cost and decide on Alert broadcasting.

Packet Format

Alive Request (AREQ)

RID_{AID}

Alive Reply (AREP)

RID_{AID}	HCP	energy level

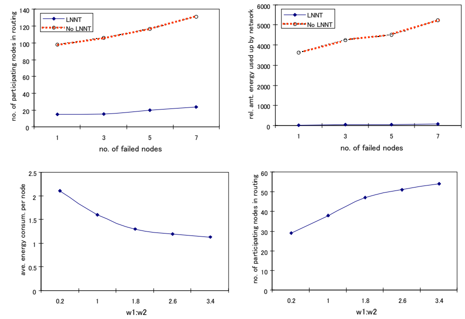
Alert Message

RID_{AID}	Failed Neighbor Nodes (RID_{AID})	HCP	energy level

Simulations

- 1260 nodes
- 1.5 m uniformly spaced
- 60 m by 60 m area
- Failed nodes located at an angle of 60 degrees relative to the sink and with a distance RID of 15-17.
- w_1, w_2, w_3 set to 1.
- In the model, each time a node broadcasts a message, the node loses *one* unit of energy.
- Two schemes simulated:
  - HexNet algorithm without LNNT
  - HexNet algorithm with LNNT

Results



Conclusions

- Proposed a new alert-based routing platform and mapping scheme for environmental field-based wireless sensor networks with the intent of reporting disaster location and scale.
- The proposed HexNet mapping and routing is aimed at disaster-monitoring environments where sensor nodes are kept at lowest cost (not equipped with localization devices such as GPS) as they are prone to inevitable destruction.
- Future work should aim at further relaxation of the strict requirements of the mapping scheme and further optimization of weight selection. Additional performance evaluation of the proposed scheme is also required.

Thank You.