A Routing Protocol for Detecting Holes in Wireless Sensor Networks with Multiple Sinks

Anju Arya
DCRUST, Murthal, Haryana, India
anjuarya12@ymail.com

Amita Malik
DCRUST, Murthal, Haryana India
amitamalik.cse@dcrustm.org

Sanjay Kumar
Hindu College of Engg. Sonepat, India
skm_09@rediffmail.com

ABSTRACT
In wireless sensor networks, data routing in an optimal way is a challenging task while tackling the limited energy constraint at the same time. Our goal is not only to enhance network's lifetime for the available limited network energy, but also to detect the sub-areas in the network, which have run out of energy or became a dead zone i.e. hole. The hole detection algorithm used in our work detects such dead zones and help the network coordinators to mitigate the problem before it hampers the whole network functioning. The simulation results have shown the ability of the protocol to optimally route data packets in a multiple sink scenario and detect hole(s) when it occurs. Our work on holes covers coverage hole detection and routing hole detection.

Categories and Subject Descriptors
• Networks → Network types → Cyber-physical networks → Sensor networks

Keywords
Reinforcement Learning (RL); Wireless Sensor Network (WSN); Routing Protocols; Q learning; Holes.

1. INTRODUCTION
With the technological advancements in sensors technology i.e. MEMS, the interest in wireless sensor networks has increased tremendously. A wireless sensor network (WSN) [1] is a collection of autonomous sensors densely deployed in a structured or unstructured manner to monitor physical environment and collect relevant information and cooperatively transfer it through the network to the sink via intermediate sensor nodes. The sensor networks have found a major application area in monitoring activities or networks. A number of routing protocols have been developed for sensor networks. These protocols try to achieve or handle energy efficiency, fault tolerance, mobility, holes problem, QoS, or combinations of above.

The ultimate goal is to maximize network lifetime with efficient routing abilities such as minimum delay, failure handling, avoiding energy wastage, etc. So much work has been done by the researchers at both the MAC layer and the network layer to achieve this. Thus, resource management is of critical importance to wireless sensor networks.

WCI '15, August 10 - 13, 2015, Kochi, India

DOI: http://dx.doi.org/10.1145/2791405.2791480

In this paper, we present a multicast routing protocol based on reinforcement learning approach. The salient features of our protocol includes optimal routing, sink mobility support, failure handling and recovery, and hole detection. This paper further describes the target application scenario, overview, problem definition, and implementation details of our protocol in the sections given below.

2. RELATED EFFORTS
A large number of routing protocols have been developed for wireless sensor networks, but there is no general protocol which could cover most of the application scenarios. The available routing protocols are specialized to fit into the required application scenario. Various multicast routing protocols and protocols supporting sink mobility have been developed from MANETs protocols for multicasting and handling mobility. But, these protocols are less energy and memory efficient as compared to reinforcement learning based protocols, which are quite effective for learning network dynamics.

Table 1: Protocols based on RL approach

Protocol	Model/ Concept used	Key features
FROMS [2,3]	Q-routing	MSR, sink mobility, fault tolerant
CLIQUE [4]	Q-learning	MSR, sink mobility, fault tolerant, EAR
QAZP [5]	Q-learning	MSR, EAER
InRout [6]	Q-learning	MSR, EAER
ATP [7]	adaptive spanning trees & constraint-based routing	sink mobility, fault tolerant
HLETDR [8]	Combination of statistics & RL	sink mobility, Fault-tolerant routing
MO-IAR[9]	ant-based	Fault-tolerant routing
IDR [10]	Real-Time Reinforcement Learning	Location based routing
Q-PR [11]	Bayesian decision model	Location based routing, EAR
RLGR [12]	Routing problem considered as MDP-solved through RL.	Location based routing, EAR

QoS routing [14]	deadline-driven dual-path routing scheme	EAR
FEAR[15]	fuzzy tech	EAER
DRLR [16]	Learning Automata	EAER

3. PROBLEM DEFINITION & MODELING

In this section, we will model the multicast routing problem in multiple sink scenarios as a reinforcement learning based problem and brief the holes problem again.

3.1 Problem Definition

Consider a sensor network in the form of a graph $G(V,E)$, where V represents a set of vertices and $\mathbf{E}$ represents a set of edges. Each vertex V_i represent a sensor node and each edge e_{ij} represent a bidirectional connection between nodes V_i and V_j [17]. Now, consider a set of source nodes S_i and a set of sink or destination nodes D_i, where S_i, D_i belongs to V. We define optimal path routing from a source S_i to multiple sinks as a minimum cost path originating from source S_i and ending or reaching each sink D_i. This path takes the form of a spanning tree $P(V_i, E_i)$ whose root is S, leaf nodes are D_i's, and rest of the vertices are intermediate sensor nodes or hops. The net cost of this optimal path or tree is defined as a function of spanning tree $C(P)$. The total cost includes the cost of transmissions and computations at each node.

3.2 Modelling Routing Problem as RL Problem

The network learning problem of our multicast routing protocol can be modeled as reinforcement learning problem. We are using a model free based RL approach, which is primarily concerned with how to obtain an optimal policy when a model is not known in advance. The RL technique used is Q learning, which is a generalization technique or a temporal difference learning based technique.

Agent: Each sensor node can be considered as a RL agent, whose current value or nodes state is the mapping between sink IDs and the hop costs to reach those sinks.

The current routing state is the pair of destination IDs (subset of sinks) and the routing information available to reach those destination IDs.

$< D_i$, *neighbor info associated with* $D_i's >$

Where $D_i \subseteq D$

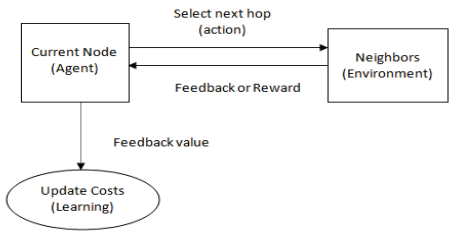

Figure 1: Modeling routing problem as RL problem

Action: The action is the selection of next hops for routing the data packets. Generally, in Q-learning an action is single new state, which is a sensor node in our case. But, in our routing protocol, an action may or may not consist of more than one states or neighbors selected for data routing. In other words, an action may be a combination of multiple single sub-actions, where each sub-action is a single new state or neighbor just like in actual or basic Q-learning method.

Policy Function: It is defined as the mapping from a state to an action. In our model, it is done through computation. An action is a route, which contains the list of next hops for routing. The current state (i.e. Node) is mapped to an action (i.e. route) by computing the Q-values of each possible route, which is actually the sum of Q-values associated with each neighbor selected as next hop for routing.

$$Q_{route} = \sum_{i=0}^{N} Q_i \quad (1)$$

Where N represents no. of next hops and Q_i is the Q-value of $next\_hop_i$ for a sink combination.

The route with the smallest Q-value possible is selected as an action for mapping with the current state.

Q-values: The Q-value in our case represents the hop cost. For each neighbor, the Q-value is calculated for different combinations of sinks reachable from this neighbor. For example, consider a neighbor n and D_i represents the sinks reachable from neighbor n. Let C_i represent a combination or subset of D_i.

$$C_i = \{ D_0, D_1, \ldots\ldots D_m \} \quad (2)$$

Then Q-value for neighbor n for a particular combination C_i is given by

$$D_{nC_i} = \quad Q(C_i) = f(HC, RE) \quad (3)$$

Where HC is the hop count and RE is the residual energy or battery of node N.

When a neighbor n is evaluated as a next hop for a route, then $C_i \subseteq D_i$ must be true.

Q-value Updation: The Q-value associated with a neighbor n for a particular sink combination is updated when a feedback is received from a neighbor after sending the data packet to it.

$$D_{new}(C_i) = Q_{calculated}(C_i) + \alpha * diff \quad (4)$$

Where $diff = feedbackvalue - current\_hop_{cost}(C_i)$ and it also represents the quality of decision as discussed below, C_i represents a sink combination, and α represents learning rate. $Q_{calculated}(C_i)$ is the calculated by equation (7) given in section 6.

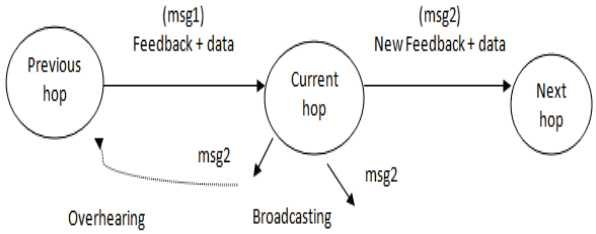

Figure 2: Broadcasting data packet piggybacked with feedback data

The feedback helps in evaluating the quality of decision taken by the current node by selecting the neighbor n as a next hop from which the feedback is received. There are three possible cases.

Case 1: Positive feedback

When $feedback < current\_hop_{cost}(C_i)$, where C_i is the sink combination for which feedback is received.

Result: Bad decision

Case 2: Negative feedback
When $feedback > current\_hop_{cost}(C_i)$

Result: Good decision

Case 3: Neutral feedback
When $feedback = current\_hop_{cost}(C_i)$

Result: Neither good nor bad i.e. fair decision

The current hop cost for C_i is updated in all the cases mentioned above. This is the actual learning done in terms of hop costs by the nodes through Q-learning approach.

Reward or Feedback value: The feedback is equal to the calculated hop cost for the set of sinks or destinations (D) by the next hop plus one. One is added to the calculated hop cost because this value would be directly used for comparison or evaluation by the current hop, which is previous hop for selected next hop, when it overhears the message broadcasted by the next hop.

3.3 Holes problem

The holes problem is the problem of detecting a group of non-functional or dead nodes in the network. The main reasons for hole creation in the network are sensors failure, existence of obstacles such as areas or conditions unfavorable to sensor operation, and low density regions due to non-uniform deployment. Generally, there are four types of holes namely, coverage holes, routing holes, jamming holes, and black/sink holes, described in [2]. In our work, we are concentrating on coverage holes and routing holes. Holes affects the network performance considerably, especially when it is large enough to cause a major section or area of the network field remain un-monitored and result in inaccurate and incomplete data gathering and analysis. Hole detection is generally accompanied by network boundary detection and hole boundary detection. The network boundary detection [3] is not our concern. We have assumed that network boundary or size is known already. In our work, we have focused on the hole boundary and the dead nodes residing on that boundary. This information helps to find the hole size and the impact, it can have on the network performance.

In our protocol, we are detecting holes by applying some tests or conditions on the nodes declared as dead or faulty. Actually, we run two algorithms for this purpose namely, coverage hole detection algorithm and routing hole detection algorithm. The priori information needed for this purpose is the location of all the nodes in the network and the knowledge about the neighboring nodes of all the sensor nodes. After detecting a hole, we can categorize the hole according to the state of nodes forming this hole.

4. PROPOSED PROTOCOL

The working of our routing protocol can be explained in two phases namely, optimal multicast routing and hole detection.

4.1 Phase I: Optimal multicast routing

The first phase is finding optimal paths available for routing data packets from their source to all interested sinks. Consider a network shown in figure 3 with one source S, two sinks (S1, S2) and one base station (BS). Suppose, source S received a data request from sink S1 and now it is ready to send the data. There are many possible paths to send data from S to S1. The shortest path consists of 3 intermediate hops, which may be <1, 3, 6>, <1, 4, 6>, or <1, 4, 7>. The challenge is to identify the best optimal path for routing. The path can be optimal in terms of minimum delay, minimum hop count, maximum residual energy or combinations of above factors. Our protocol will find this optimal path through the local information available about the hops to be traversed through a neighbor, which is available through updates or feedbacks.

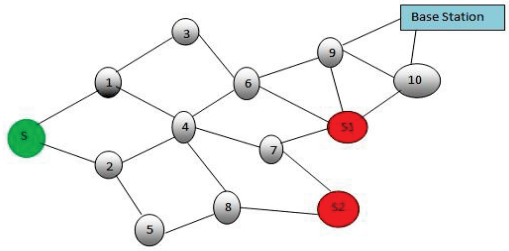

Figure 3: A sample network with 2 sinks, 1 source, and 1 base station

Next Hop Selection: The initial routing information for all the nodes is obtained through initial sink and base station announcements, which is done by broadcasting DATA_REQ messages by them. The gathered information is updated through feedbacks obtained after sending data packets to the neighbors and through periodic data requests messages. The periodic data requests messages also indicate that the sink is still interested in data and alive too. The frequency of sending these requests by sinks is much more than the base station because the sinks can be mobile or static, so they need to update their current location and interest, in case they move out of the network, as compared to the base station, which is static and needs to be updated only when either a new neighbor is discovered or a neighbor is deleted (in case of death) by the nodes in the network.

The nodes chooses the next hop for routing the data packet based on the local information collected about the neighbors of their relative distance or cost to reach the sink. The next hop is chosen not just on the basis of minimum hop count, but also on the basis of other factors like the total energy required for transmission. The selected next hop or neighbor is more optimal if it is able to share some next hops for some sink pairs as compared to the neighbors having low hop count than the selected next hop. The neighbor capable of sharing one or more next hops saves some extra energy, which might have wasted by sending the data packet through different neighbors separately for the same set of sinks with low or same hop count. Thus, the routing protocol needs to explore other non optimal routes too with the hope of getting more low cost routes than the available ones. Consider an example of sending data to both the sinks (S1, S2) from source S. For (S, S1) source-destination pair, some possible paths are <S, 1, 3, 6, S1>, <S, 1, 4, 7, S1>, <S, 1, 4, 6>, <S, 2, 4, 7, S1>, etc. And for (S, S2) pair, some possible paths are <S, 1, 4, 7, S2>, <S, 2, 5, 8, S2>, <S, 2, 4, 8>, <S, 2, 4, 7, S2>, etc. This protocol can select 2 or 1 as next hop for both the sinks since the hop cost is low for both S1 and S2 from S and also they share a common next hop and may also share some more hops further, which is useful for saving some network energy. The decision for selecting next hop is taken by selecting the route with lowest or optimal Q-value possible, which depends on both hop cost and battery cost factors.

Decision Evaluation: The quality or goodness of decision taken can be evaluated through the feedback obtained after routing data packet to the next hop. The next hop piggybacks its cost estimations with the data of the message packet and broadcast it. The neighbors (including previous hop or sender) of this next hop, except the neighbor selected for re-routing the piggybacked data packet overhears the broadcasted message and extract the piggybacked data, also called feedback, to improve their costs

estimations of the neighbors or next hops, which actually equal real hop costs for routes.

Improving Hop Cost Estimations of Neighbors: The source sends data packets to all its neighbors a several number of times until its cost estimations converge or the cost values stabilize. This process is also repeated by the neighbors to stabilize their cost estimations for their neighbors. All this is part of the exploration process. The feedback mechanism also helps in handling recovery, mobility, and failures by registering new routes for new or recovered nodes, deleting entries for dead or moved nodes and updating base station about the dead or faulty nodes in case of failures.

4.2 Phase II: Hole Detection

During routing, if any node is about to die, then it broadcast DEATH message to all its neighbors.

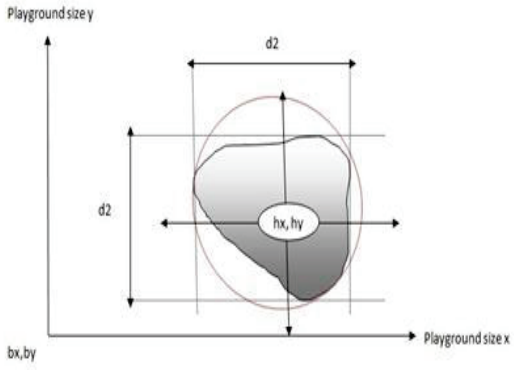

Figure 4: A plotted hole area on a graph

The neighboring nodes receiving this message deletes the dead node from their neighbors table and also sends the coordinates of the dead neighbor node to the base station in the form of "Hole info" data.

The base station runs hole detection algorithm periodically on the network and use dead nodes information available for hole detection. If a hole is detected, then the base station can send information about the coordinates of the dead nodes on the hole boundary along with the type of hole detected to the main controller, if present.

< deadnode_coord_list, type>

The protocol also gives other information regarding the hole area such as center of the ellipsoidal area enclosing the hole area and the diameters along the x-axis and y-axis of the ellipse, see figure4.

The above information is useful for deployment of both uniform type (in case of holes consisting of dead nodes or no nodes) and non-uniform type (along the perimeter for dead rings).

5. EXPLORATION

The exploration strategy helps to discover new optimal routes in the network and helps in the selection of best possible optimal routes. In our protocol, we have used epsilon greedy strategy to explore new optimal routes. It is a greedy algorithm, which exploits the best available path and at the same time explores other paths too. There are some more strategies available for exploration such as uniform explore, stochastic weighted explore, weighted explore, etc. The exploration strategy chosen must be in accordance with the protocol requirements or objectives.

6. COST FUNCTION

The Q-values associated with different sink combinations for a neighbor are calculated using a cost function. The cost function in our protocol uses hop count and residual energy of the node as the cost metrics for calculating Q-values.

Suppose, we want to calculate Q-value for a route R at node N. Then,

$$Q(R) = f(HC, RE) \quad (5)$$

Where *HC* is the hop count and *RE* is the residual energy or battery of node *N*.

$$Q_{route} = \sum_{i=0}^{N} Q(C_i) \quad (6)$$

Where N represents no. of next hops and $Q(C_i)$ is the Q-value of next_hop$_i$ for a sink combination C_i.

$$Q(R) = \beta_{RE} - \lambda_{HC} \quad (7)$$

$$Or\ Q(C_i) = \beta_{RE} - hop\_count_{actual} \quad (8)$$

Equation 8 can be used when negative Q values are not allowed. The values of β_{RE} and λ_{HC} varies according to the maximum hop count and the residual energy or battery of node N.

$$\beta_{RE} = f(residual\ energy) = 1 - residual\ energy \quad (9)$$

In terms of percentage,

$$\beta_{RE} = [1 - remaining\ energy\%] * total\ energy \quad (10)$$

The less is the remaining energy, the more value it will add to the Q-value of the route R making it less favourable to selection.

$$\lambda_{HC} = f(hop\_count_{max}, hop\_count_{actual}) \quad (11)$$

$$hop\_countactual = (\sum_{0}^{m} hopcount(D_i)) - m + 1 \quad (12)$$

Where hop_count_{max} is the maximum hop count allowed and hop_count_{actual} is the hop count required from the current node N.

$$\lambda_{HC} = hop\_count_{max}, hop\_count_{actual} \quad (13)$$

Thus, the more is the difference between the above values, the more it reduces the Q-value of the route R, making it more favourable to selection. In case hop_count_{max} is less than hop_count_{actual}, then this factor will increase the Q-value making the route R less favourable to selection.

While evaluating a route R for optimality, we first check whether it conform to the major two requirements of optimality i.e. $hop\_count_{actual} < max.hop\ count$ and $RE_R > min.battery$, where RE is the residual energy of node N. Then, among the routes conforming with the lowest Q-value on the basis of optimality criteria, we have three cases as follows:

Best Case: when both the optimality criteria are satisfied, then route R obtained is surely an optimal route.

Average Case: when either one of the condition is satisfied like when residual energy condition is met but hop count of available routes is more than max. hop count. Another situation is when hop count criteria is met but the residual energy of the available routes is low.

Worst Case: when none of the condition is met and we have to select among the available routes the route with lowest Q-value possible.

7. SIMULATION RESULTS

We have used omnet++, its mobility framework, and a feedback framework for simulating and evaluating our protocol. We simulate our protocol with two objectives. One objective is to study the routing performance over a simulation period and the other objective is to evaluate the ability to detect hole by our protocol. We have used Epsilon Greedy as the exploration strategy.

7.1 Routing Performance

The sample network used for analysis is a connected 30-node topology on a field size of 1500x1500m, with a maximum transmission or communication range of 30m. The data requests are sent after every 10sec by all the sinks and the sources send data after every 2sec. The maximum battery of the sensor node is 5000. The sensor nodes are assigned different battery capacities to check the network behavior. The simulation time is 450s.

Figure 5 shows the packet delivery delay at various simulation times. The graph clearly displays that delay time is increasing with time. The reasons may be packet collisions or multiple transmission attempts for some packets during that time and also due to death of some sensor nodes in the network.

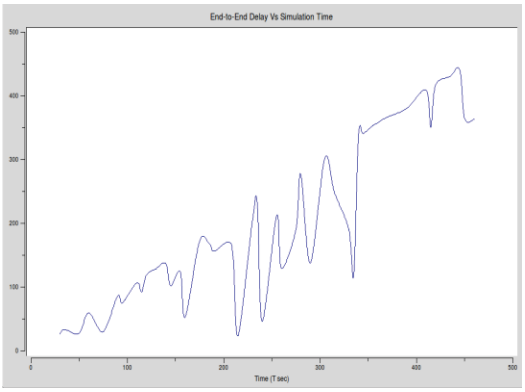

Figure 5: Packet delivery time Vs Simulation time (including only those simulation times when there are packet deliveries)

Figure 6 shows the energy distribution in the network. Initially, the distribution value is constant for some period of time, which represents a uniform distribution. After 400s of simulation period, the graph line moves downward, which can be due to non-uniform energy distribution and decrease in energy of sensor nodes.

7.2 Hole detection results

The sample network used for analysis is a connected 100-node topology on a field size of 150x150m, with a maximum transmission or communication range of 10m, see figure 9. The data requests are sent after every 10sec by all the sinks and the sources send data after every 2sec.

A hole detected can be categorized into following types based on the state of sensor nodes forming it.

Type 1. Dead nodes only
Type 2. Dead and inaccessible nodes (due to dead rings)
Type 3. No sensor node deployment

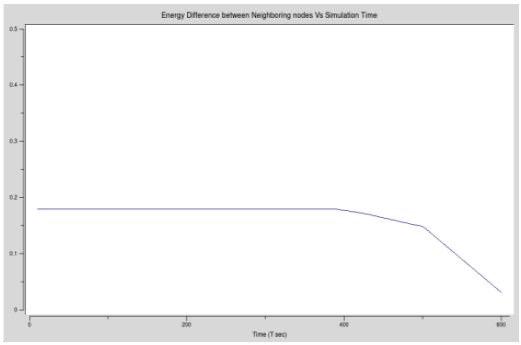

Figure 6: Energy distribution of the network

Consider the sample network shown in figure 7. Now, suppose some of the nodes are dead as shown in figure 8. In this case, a hole is detected. The area of the hole detected is shown in graph by plotting the coordinates of the dead nodes of the hole on the graph as shown in figure 9. The blackened sub-area shows the hole area covered by the dead nodes.

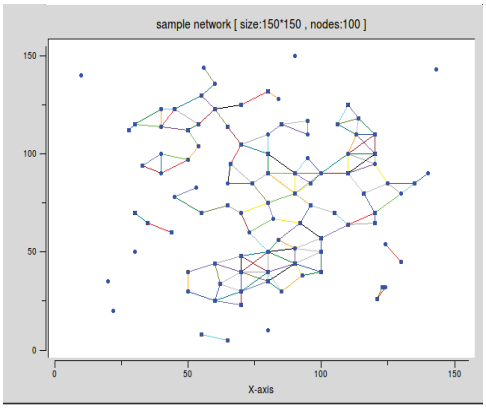

Figure 7: Sample network

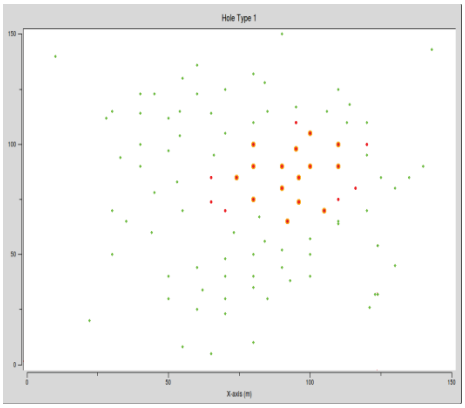

Figure 8: A Graph displaying some dead nodes (red/yellow) in sample network without connections

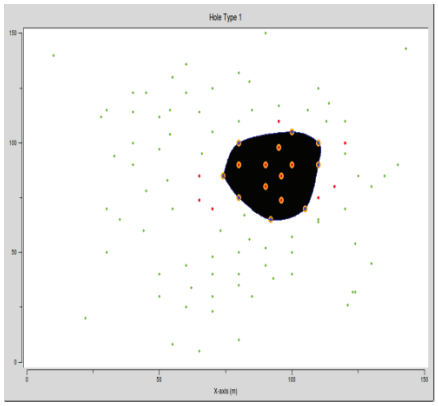

Figure 9: A Graph displaying the hole area as blackened portion

8. CONCLUSION

The reinforcement learning approach used by our protocol has proved to be effective in selecting optimal paths during routing. The simulation results have clearly demonstrated the ability of our protocol to support sink mobility, handle node's failure, and detect holes occurring in the network

9. ACKNOWLEDGEMENTS

The work presented in this paper was partially supported by the University Grants Commission under grant number 41-626/2012 (SR).

10. REFERENCES

[1] Tubaishat, M., & Madria, S. (2003). Sensor networks: an overview. *Potentials, IEEE*, *22*(2), 20-23.

[2] Villaverde, B. C., Rea, S., & Pesch, D. (2012). InRout–A QoS aware route selection algorithm for industrial wireless sensor networks. *Ad Hoc Networks*,*10*(3), 458-478.

[3] Khan, I., Mokhtar, H., & Merabti, M. (2008, June). A survey of boundary detection algorithms for sensor networks. In *Proceedings of the 9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*.

[4] Förster, A., & Murphy, A. L. (2009, June). CLIQUE: Role-free clustering with Q-learning for Wireless Sensor Networks. In *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on* (pp. 441-449). IEEE.

[5] Villaverde, B. C., Rea, S., & Pesch, D. (2012). InRout–A QoS aware route selection algorithm for industrial wireless sensor networks. *Ad Hoc Networks*,*10*(3), 458-478.

[6] Zhang, Y., & Huang, Q. (2006). A learning-based adaptive routing tree for wireless sensor networks. *Journal of Communications*, *1*(2), 12-21.

[7] Baruah, P., Urgaonkar, R., & Krishnamachari, B. (2004, November). Learning-enforced time domain routing to mobile sinks in wireless sensor fields. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on* (pp. 525-532). IEEE.

[8] GhasemAghaei, R., Rahman, A. M., Rahman, M. A., & Gueaieb, W. (2008, March). Ant colony-based many-to-one sensory data routing in wireless sensor networks. In *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on* (pp. 1005-1010). IEEE.

[9] Zhang, Y., Liu, J., & Zhao, F. (2006). Information-directed routing in sensor networks using real-time reinforcement learning. In *Combinatorial Optimization in Communication Networks* (pp. 259-288). Springer US.

[10] Arroyo-Valles, R., Alaiz-Rodríguez, R., Guerrero-Curieses, A., & Cid-Sueiro, J. (2007, December). Q-probabilistic routing in wireless sensor networks. In*Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on* (pp. 1-6). IEEE.

[11] Dong, S., Agrawal, P., & Sivalingam, K. (2007, November). Reinforcement learning based geographic routing protocol for UWB wireless sensor network. In*Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE* (pp. 652-656). IEEE.

[12] Liu, Z., & Elhanany, I. (2006, April). RL-MAC: A QoS-aware reinforcement learning based MAC protocol for wireless sensor networks. In *Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on* (pp. 768-773). IEEE.

[13] Mahapatra, A., Anand, K., & Agrawal, D. P. (2006). QoS and energy aware routing for real-time traffic in wireless sensor networks. *Computer Communications*, *29*(4), 437-445.

[14] ALMomani, I. M., & Saadeh, M. K. (2011). FEAR: Fuzzy-Based Energy Aware Routing Protocol for Wireless Sensor Networks. *Int'l J. of Communications, Network and System Sciences*, *4*(06), 403.

[15] Kordafshari, M. S., Pourkabirian, A., Meybodi, M. R., & Movaghar, A. (2012, May). Distributed QoS routing algorithm in large scale Wireless Sensor Networks. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on* (pp. 826-830). IEEE.

[16] Ahmed, N., Kanhere, S. S., & Jha, S. (2005). The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mobile Computing and Communications Review*, *9*(2), 4-18.

[17] Kumar, S., Dave, M., & Dahiya, S. (2014). ACO based QoS aware routing for wireless sensor networks with heterogeneous nodes. In *Emerging Trends in Computing and Communication* (pp. 157-168). Springer India.