DRINA: AN ENHANCEDRELIABLE AND LIGHTWEIGHT ROUTINGAPPROACH WITH ROUTE REPAIR MECHANISM FOR IN-NETWORK AGGREGATION IN WIRELESS SENSOR NETWORKS

C.P.Sameerana[1], Sarala D.V[2]
[1]H.O.D, Associate Professor, Dept.of. CSE, APSCE, Bangalore
[2]Assistant Professor, Dept. of. CSE,APSCE, Bangalore
Email: hodcse@apsce.ac.in[1], sarala5.dv@gmail.com[2]

Abstract

Large scale dense Wireless Sensor Networks (WSNs) will be increasingly deployed in different classes of applications for accurate monitoring. Due to the high density of nodes in these networks, it is likely that redundant data will be detected by nearby nodes when sensing an event. Since energy conservation is a key issue in WSNs, data fusion and aggregation should be exploited in order to save energy. In this case, redundant data can be aggregated at intermediate nodes reducing the size and number of exchanged messages and, thus, decreasing communication costs and energy consumption. In this work, we propose a novel Data Routing for In-Network Aggregation, called DRINA, that has some key aspects such as a reduced number of messages for setting up a routing tree, maximized number of overlapping routes, high aggregation rate, and reliable data aggregation and transmission. The proposed DRINA algorithm was extensively compared to two other known solutions: the Information Fusion-based Role Assignment (InFRA) and Shortest Path Tree (SPT) algorithms. Our results indicate clearly that the routing tree built by DRINA provides the best aggregation quality when compared to these other algorithms. The obtained results show that our proposed solution outperforms these solutions in different scenarios and in different key aspects required by WSNs.

Index Terms: Routing protocol, in-network aggregation, wireless sensor networks

INTRODUCTION

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous devices that cooperatively sense physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants at different locations. WSNs have been used in applications such as environmental monitoring, homeland security, critical infrastructure systems, communications, manufacturing, and many other applications that can be critical to save lives and assets. Sensor nodes are energy-constrained devices and the energy consumption is generally associated with the amount of gathered data, since communication is often the most expensive activity in terms of energy. For that reason, algorithms and protocols designed for WSNs shouldconsider the energy consumption in their conception. Moreover, WSNs are data-driven networks that usually produce a large amount of information that needs to be routed, often in a multi-hop fashion, toward a sink node, which works as a gateway to a monitoring center. Given this scenario, routing plays an important role in the data gathering process.

A possible strategy to optimize the routing task is to use the available processing capacity provided by the intermediate sensor nodes along the routing paths. This is known as data-centric routing or in-network data aggregation. For more efficient and effective data gathering with a minimum use of the limited resources, sensor nodes should be configured to smartly report data by making local decisions. For this, data aggregation is an effective technique for saving energy in WSNs. Due to the inherent redundancy in raw data gathered by the sensor nodes, in-networking aggregation can often be used to decrease the communication cost by

eliminating redundancy and forwarding only smaller aggregated information. Since minimal communication leads directly to energy savings, which extends the network lifetime, in-network data aggregation is a key technology to be supported by WSNs. In this work, the terms information fusion and data aggregation are used as synonyms. In this context, the use of information fusion is twofold: 1) to take advantage of data redundancy and increase data accuracy, and 2) to reduce communication load and save energy. One of the main challenges in routing algorithms for WSNs is how to guarantee the delivery of the sensed data even in the presence of nodes failures and interruptions in communications. These failures become even more criticalwhen data aggregation is performed along the routing paths since packets with aggregated data contain information from various sources and, whenever one of these packets is lost a considerable amount of information will also be lost. In the context of WSN, data aggregation awarerouting protocols should present some desirable characteristics such as: a reduced number of messages for setting up a routing tree, maximized number of overlapping routes, high aggregation rate, and also a reliable data transmission.

EXISTING SYSTEM

In most cases, tree-based protocols build a traditional shortest path routing tree. For instance, the Shortest Path Tree (SPT) algorithm uses a very simple strategy to build a routing tree in a distributed fashion. In this approach, every node that detects an event reports its collected information by using a shortest path to the sink node. Information fusion occurs whenever paths overlap (opportunistic information fusion).

SPT (Shortest Path Tree) is a commonly used topology in WSNs as each sensor node in a SPT reaches the root with the smallest number of hops. However, a randomly constructed SPT may not increase network lifetime. In new weighted path cost function improving the SPT approach, each link is assigned a weight according to its path length to the root, and a link closer to the root has a larger weight. By balancing load according to the links' weights, this approach increases network lifetime compared with those randomly constructed SPT. For the problem of finding a maximum lifetime tree from all the shortest path trees in a WSN, They first build a fat tree which contains all the shortest path trees. Then, they propose a method based on each node's number of children and its initial energy to find a minimum load shortest path tree to convert the problem into a semi-matching problem, and solve it by the min-cost maximum flow approach in polynomial time. Proposes an approximation algorithm for maximizing network lifetime by constructing a min-max-weight spanning tree, which guarantees thebottleneck nodes having the least number of descendants. The approximation algorithm iteratively transfers some of the descendants of the nodes with the largest weight to the nodes with smaller weights.

Similarly to the tree-based approaches, cluster-based schemes also consist of a hierarchical organization of the network.The Information Fusion-based Role Assignment (InFRA) algorithm builds a cluster for each event including only those nodes that were able to detect it. Then, cluster-heads merge the data within the cluster and send the result toward the sink node. The InFRA algorithm aims at building the shortest path tree that maximizes information fusion. Thus, once clusters are formed, cluster-heads choose the shortest path to the sink node that also maximizes information fusion by using the aggregated coordinators distance. A disadvantage of the InFRA algorithm is that for each new event that arises in the network, the information about the event must be flooded throughout the network to inform other nodes about its occurrence and to update the aggregated coordinators-distance. This procedure increases the communication cost of the algorithm and, thus, limits its scalability.

PROPOSED SYSTEM

The main goal of our proposed the DRINA algorithm is to build a routing tree with the shortest paths that connect all source nodes to the sink while maximizing data aggregation. The proposed algorithm considers the following roles in the routing infrastructure creation:

- Collaborator- A node that detects an event and reports the gathered data to a coordinator node.
- Coordinator- A node that also detects an event and is responsible for gathering all the gathered data sent by collaborator nodes, aggregating

them and sending the result toward the sink node.

- Sink- A node interested in receiving data from a set of coordinator and collaborator nodes.
- Relay- A node that forwards data toward the sink.

The DRINA algorithm can be divided into three phases. In Phase 1, the hop tree from the sensor nodes to the sink node is built. In this phase, the sink node starts building the hop tree that will be used by Coordinators for data forwarding purposes. Phase 2 consists of cluster formation and cluster-head election among the nodes that detected the occurrence of a new event in the network. Finally, Phase 3 is responsible for both setting up a new route for the reliable delivering of packets and updating the hop tree.

Phase 1: Building the Hop Tree

In this phase, the distance from the sink to each node is computed in hops. This phase is started by the sink node sending, by means of a flooding, the Hop Configuration Message (HCM) to all network nodes. The HCM message contains two fields: ID and HopToTree, where ID is node identifier that started or retransmitted the HCM message and HopToTree is the distance, in hops, by which an HCM message has passed. The HopToTree value is started with value 1 at the sink, which forwards it to its neighbors (at the beginning, all nodes set the HopToTree as infinity). Each node, upon receiving the message HCM, verifies if the value of HopToTree in the HCM message is less than the value of HopToTree that it has stored and if the value of FirstSending is true. If that condition is true then the node updates the value of the NextHop variable with the value of the field ID of message HCM, as well as the value of the HopToTree variable, and the values in the fields ID and HopToTree of the HCM message. The node also relays the HCM message. Otherwise, if that condition is false, which means that the node already received the HCM by a shorted distance, then the node discards the received HCM message. The steps described above occur repeatedly until the whole network is configured.Before the first event takes place, there is no established route and the HopToTree variable stores the smallest distance to the sink. On the first event occurrence, HopToTree will still be the smallest distance; however, a new route will be established. After the first event, the HopToTree stores the smaller of two values: the distance to the sink or the distance to the closest already established route.

Algorithm 1:

Step1: Sink node sends a broadcast of HCM message with a value of HopToTree=1.

Step2: Check if HopToTree value of node is less than the value which it has stored and the value of FirstSending is TRUE.

Step3: If condition is true then the node updates the value of next hop variable with the value of the field ID of HCM message as well as the value of HopToTree variable and the values in the fields ID.

Step4: If that condition is false, then the node before the first event takes place, there is no established route and the HopToTree variable stores the smallest distance to the sink.

Step5: On the first event occurrence HopToTree will be the smallest distance.

Step6: After the first event the HopToTree stores the smallest of two values: the distance to the sink or the distance to the closest already established route.

Phase 2: Cluster Formation

When an event is detected by one or more nodes, the leaderelection algorithm starts and sensing nodes will be runningfor leadership (group coordinator); this process is describedin Algorithm 2. For this election, all sensing nodes areeligible. If this is the first event, the leader node will be theone that is closest to the sink node. Otherwise, the leaderwill be the node that is closest to an already establishedroute. In the case of a tie, i.e., twoor more concurrent nodes have the same distance in hops tothe sink (or to an established route), the node with thesmallest ID maintains eligibility. Another possibility is to use the energylevel as a tiebreak criterion.At the end of the election algorithm only one node inthe group will be declared as the leader (Coordinator). Theremaining nodes that detected the same event will be theCollaborators. The Coordinator gathers the informationcollected by the Collaborators and sends them to the sink. Akey advantage of this algorithm is that all of the informationgathered by the nodes sensing the same event will beaggregated at a single node (the Coordinator), which ismore efficient than other aggregation mechanisms (e.g, opportunistic aggregation).

Algorithm 2:

Step1: If this is the first event, the leader node will be the one that is closest to the sink node otherwise the leader will be the node that is closest to an already established route.

Step2: If two or more concurrent nodes have the same distance in hop to the sink the node with the smallest ID maintaince eligibility.

Step3: If still there exists tie, then energy level of nodes is used as tie break.

Phase 3: Routing Formation and Hop Tree Updates

The elected group leader, as described in Algorithm 2, starts establishing the new route for the event dissemination. This process is described in Algorithm 3. For that, the Coordinator sends a route establishment message to its NextHop node. When the NextHop node receives a route establishment message, it retransmits the message to its NextHop and starts the hop tree updating process. These steps are repeated until either the sink is reached or a node that is part of an already established route is found. The routes are created by choosing the best neighbor at each hop. The choices for the best neighbor are twofold: 1) when the first event occurs, the node that leads to the shortest path to the sink is chosen; and 2) after the occurrence of subsequent events, the best neighbor is the one that leads to the closest node that is already part of an established route. This process tends to increase the aggregation points, ensuring that they occur as close as possible to the events. The resulting route is a tree that connects the Coordinator nodes to the sink. When the route is established, the hop tree updating phase is started. The main goal of this phase is to update the HopToTree value of all nodes so they can take into consideration the newly established route. This is done by the new relay nodes that are part of an established route. These nodes send an HCM message (by means of a controlled flooding) for the hop updating. The whole cost of this process is the same of a flooding, i.e., each node will send only one packet. This algorithm for the hop updating follows the same principles of the hop tree building algorithm.

Algorithm 3:

Step1: The Coordinator sends a route establishment message to its NextHop node. When the NextHop node receives a route establishment message, it retransmits the message to its NextHop and starts the hop tree updating process.

Step2: These steps are repeated until either the sink is reached or a node that is part of an already established route is found.

Step3: The routes are created by choosing the best neighbor at each hop. The choices for the best neighbor are twofold:

1) When the first event occurs, the node that leads to the shortest path to the sink is chosen and

2) After the occurrence of subsequent events, the best neighbor is the one that leads to the closest node that is already part of an established route.

Step4: When the route is established, the hop tree updating phase is started.

While the node has data to transmit,it verifies whether it has more than one descendant thatrelays its data. If it is the case, it waitsfor a period of time and aggregates all data received andsends the aggregated data to its NextHop. Otherwise, it forwards the data to its NextHop.For every packet transmission with aggregated data, theRoute Repair Mechanism is executed as shown in Algorithm 3. A route repair mechanism is used to sendinformation in a reliable way. Sender nodes wait a predefinedtime period to receive a packet delivery confirmation.When the confirmation is not received by the sendernode, a new destination node is selected and the message isretransmitted by that node.

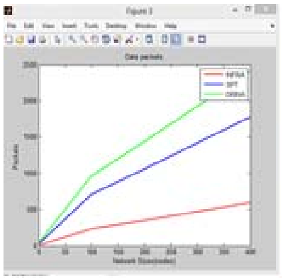
ROUTE REPAIR MECHANISM

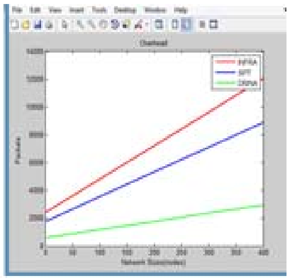
The route created to send the data toward the sink node is unique and efficient since it maximizes the points of aggregation and, consequently, the information fusion. However, because this route is unique, any failure in one of its nodes will cause disruption, preventing the delivery of several gathered event data. Possible causes of failure include low energy, physical destruction, and communication blockage. Some fault-tolerant algorithms for WSNs have been proposed in the literature. Some are based on periodic flooding mechanisms and rooted at the sink, to repair broken paths and to discover new routes to forward traffic around faulty nodes. This mechanism is not satisfactory in terms of energy saving because it wastes a lot of energy with repairing messages. Furthermore, during the network flooding period, these algorithms are unable to route data around failed nodes, causing data losses. Our DRINA algorithm offers a piggybacked, ACK-based, route repair mechanism, which consists of two parts: failure detection at the NextHop node, and selection of a new NextHop. When a relay node needs to forward data to its NextHop node, it simply sends the data packet, sets a timeout, and waits for the retransmission of the data packet by its NextHop. This re-transmission is also considered an ACK message. If the sender receives its ACK from the NextHop node, it can infer that the NextHop node is alive and, for now, everything is ok. However, if the sender node does not receive the ACK from the NextHop node within the predetermined timeout, it considers this node as offline and another one should be selected as the new NextHop node. For this, the sender chooses the neighbor with the lowest hop-to-tree level to be its new NextHop; in case of a tie, it chooses the neighbor with the highest energy level. After that, the sender updates its routing table to facilitate the forwarding of subsequent packets. After the repairing mechanism is applied, a newly partial reconstructed path is created

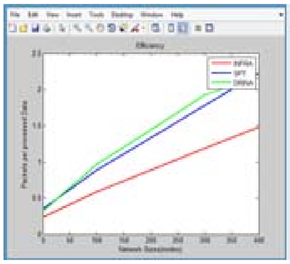
PERFORMANCE EVALUATION

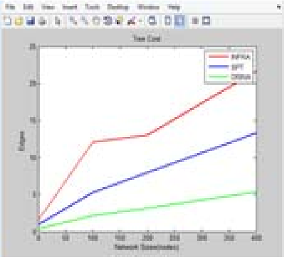
In this section, we evaluate the proposed DRINA algorithm and compare its performance to two other known routing protocols: the InFRA and SPT algorithms. These two algorithms were chosen for being well known in the literature and have the same goals that the proposed DRINA algorithm. Table 2 shows the basic characteristics of SPT, InFRA, and DRINA algorithms. We evaluate the DRINA performance under the following metrics:

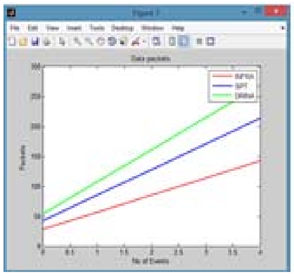
1. Packet delivery rate.
2. Control overhead.
3. Efficiency (packets per processed data).
4. Routing tree cost.
5. Loss of raw data.
6. Loss of aggregated data.
7. Transmissions number.

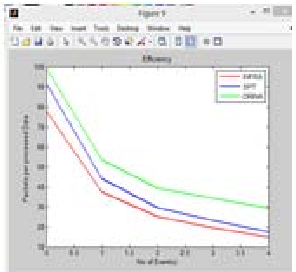


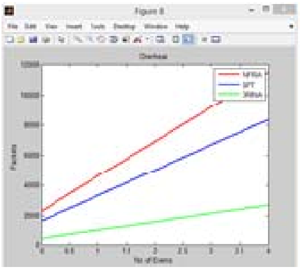


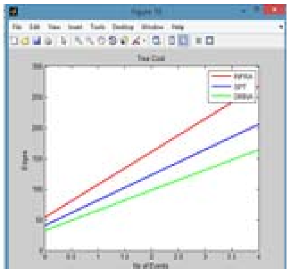








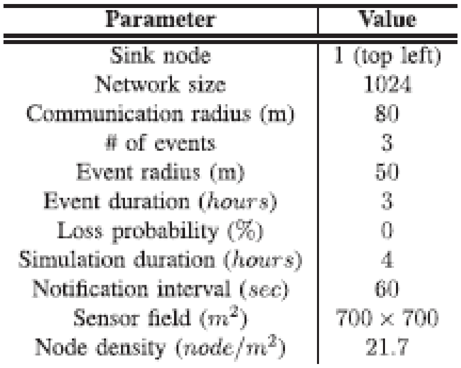




METHODOLOGY

The performance evaluation is achieved through simulations using the SinalGo version v.0.75.3 network simulator. In all results, curves represent average values, while error bars represent confidence intervals for 95 percent of confidence from 33 different instances (seeds). The default simulation parameters are presented in Table 3. For each simulation set, a parameter shown in Table 3 will be varied as described in the evaluated scenario. The first event starts at time 1,000 s and all other events start at a uniformly distributed random time between the interval ½1;000; 3;000_ seconds. Also, these events occur at random positions. The network density is considered as the relation nπr^2_c=A, where n is number of nodes, rc is the communication radius, and A is the area of the sensor field. For each simulation in which the number of nodes is varied, the sensor field dimension is adjusted accordingly in order to maintain the node density at the same value. Sensor nodes are uniformly and randomly deployed.

TABLE 3
Simulation Parameters



Parameter	Value
Sink node	1 (top left)
Network size	1024
Communication radius (m)	80
# of events	3
Event radius (m)	50
Event duration (*hours*)	3
Loss probability (%)	0
Simulation duration (*hours*)	4
Notification interval (*sec*)	60
Sensor field (m^2)	700 × 700
Node density ($node/m^2$)	21.7

To provide a lower bound to the packet transmissions, an aggregation function was used that receives p data packets and sends only a fixed size merged packet. However, any other aggregation function can be used to take advantage of DRINA features. This function is performed at the aggregation points whenever these nodes send a packet. The evaluated algorithms used periodic simple aggregation strategy in which the aggregator nodes transmit periodically the received and aggregated information. The following metrics were used for the performance evaluation:

Data packet delivery rate- Number of packets that reach the sink node. This metric indicates the quality of the routing tree built by the algorithms—the lower the packet delivery rate, the greater the aggregation rate of the built tree.

Control packet overhead- Number of control messages used to build the routing tree including the overhead to both create the clusters and set up all the routing parameters for each algorithm.

Efficiency- Packets per processed data. It is the rate between the total packets transmitted (data and control packets) and the number of data received by the sink.

Routing tree cost- Total number of edges in the routing tree structure built by the algorithm.

Loss of aggregated data- Number of aggregated data packets lost during the routing. In this metric, if a packet contains X aggregated packets and if this packet is lost, it is accounted the loss of X packets.

Number of transmissions- Sum of control overhead and data transmissions, i.e., the total packets transmitted.

Number of Steiner nodes- Number of Steiner nodes in the routing structure, i.e., the number of relay nodes.

CONCLUSION AND FUTURE WORK:

Aggregation aware routing algorithms play an important role in event-based WSNs. In this work, we presented the DRINA algorithm, a novel and reliable Data Aggregation Aware Routing Protocol for WSNs. Our proposed DRINA algorithm was extensively compared to two other known routing algorithms, the InFRA and SPT, regarding scalability, communication costs, delivery efficiency, aggregation rate, and aggregated data delivery rate. By maximizing the aggregation points and offering a fault tolerant mechanism to improve delivery rate, the obtained results clearly show that DRINA outperformed the InFRA and SPT algorithms for all evaluated scenarios. Also, we show that our proposed algorithm has some key aspects required by WSNs aggregation aware routing algorithms such as a reduced number of messages for setting up a routing tree, maximized number of overlapping routes, high aggregation rate, and reliable data aggregation and transmission.

As future work, spatial and temporal correlation of the aggregated data will also be taken into consideration as well as the construction of a routing tree that meets application needs. We also plan to modify the DRINA algorithm to stochastically select nodes that will be part of the communication structure. The goal is to find a balance between the overhead and the quality of the routing tree. In addition, new strategies will be devised to control the waiting time for aggregator nodes based on two criteria: average distance of the event coordinators, and spatial and semantics-event-correlation

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci,“Wireless

Sensor Networks: A Survey," Computer Networks, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] K. Romer and F. Mattern, "The Design Space of Wireless Sensor Networks," IEEE Wireless Comm., vol. 11, no. 6, pp. 54-61, Dec. 2004.

[3] G. Anastasi, M. Conti, M. Francesco, and A. Passarella, "Energy Conservation in Wireless Sensor Networks: A Survey," Ad Hoc Networks, vol. 7, no. 3, pp. 537-568, http://dx.doi.org/10.1016/j.adhoc.2008.06.003, May 2009.

[4] A. Boukerche, R.B. Araujo, and L. Villas, "Optimal Route Selection for Highly Dynamic Wireless Sensor and Actor Networks Environment," Proc. 10th ACM Symp. Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07), pp. 21-27,2007.

[5] O. Younis, M. Krunz, and S. Ramasubramanina, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," IEEE Network, vol. 20, no. 3, pp. 20-25, Dec. 2006.

[6] S. Olariu, Q. Xu, and A. Zomaya, "An Energy-Efficient Self- Organization Protocol for Wireless Sensor Networks," Proc. IEEE Intelligent Sensors, Sensor Networks and Information Processing Conf. (ISSNIP), pp. 55-60, Dec. 2004.

[7] H.S. AbdelSalam and S. Olariu, "A Lightweight Skeleton Construction Algorithm for Self-Organizing Sensor Networks," Proc. IEEE Int'l Conf. Comm. (ICC),pp.1-5,http://dblp.uni-trier