A Novel Data Routing For In-Network Aggregation in Wireless Sensor Network

A. V. Kalpana[1] S. Prashanthi[2] A. Vidhya[3] A. Nivetha[4]
[1]Assistant Professor [2,3,4] U.G. Student
[1,2,3,4]Department of Computer Science & Engineering
[1,2,3,4] Velammal institute of technology, panchetti.

Abstract— Wireless sensor network will be increasingly deployed in several application for accurate monitoring. In Wireless sensor network, energy conservation is one of the main problems, since wireless sensor network is used in much application such as military appliance and home monitoring etc. We need to reduce energy used by sensor node to increase the network lifetime and decrease communication cost. Due to high density of node, the nearby node will detect data while sensing an event. In order to save energy Data fusion and aggregation technique is used. In this case redundant data is aggregated at intermediate node ant it will send to the sink node. Security plays a fundamental role in Wireless sensor network. We propose a Novel Data Routing for In-network Aggregation using DRINA Algorithm and implement privacy preserving technique to achieve confidentiality and authentication. Drina algorithm is cluster-based approach compared with two existing algorithm LEACH, SPT and INFRA. Drina provides the best aggregation rate compared to existing algorithm.

Keywords: Routing protocol, Aggregation, Privacy, cluster, Wireless sensor network.

I. INTRODUCTION

A wireless sensor network (WSN) of spatially distributed autonomous sensors to Monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location(Sink)[1]. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on[2].

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. A Sensor Newark is energy-constrained device and energy consumption is generally associated with the amount of gathered data, since communication is the most expensive in term of energy. For that reason algorithm and protocol designed must consider the energy consumption [3] [4] [5].

II. EXISTING SYSTEM

WSN is a data-driven network and produce large amount of information that need to be routed in a Multihop 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.

1) 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.
2) For more efficient and effective data gathering with a minimum use of the limited resources, sensor nodes should be configured t0 smartly report data by making local decisions[6][7][8][9].

III. CHALLENGES

The challenge we need to face while designing a routing algorithm is guarantee of delivery the sensed data to sink node. The worst situations like occurrence of node broken or interruption in communication will affect the guarantee of sensed data delivery to sink node. Because in data aggregated wireless sensor network if a node failure occur then the information from various sensor nodes are lost. In wireless sensor network, security is an important aspect but there will be no privacy if any of the third parties is attacked.

IV. PROPOSED SYSTEM

The proposed scheme to Aggregation aware routing algorithms play an important role in event based WSN's the DRINA algorithm [1], a novel and reliable Data Aggregation Aware Routing Protocol for WSNs. In proposed DRINA algorithm was extensively compared to two other known routing algorithms [2], the In FRA and SPT regarding scalability, communication costs, delivery efficiency, aggregation rate and aggregated data delivery rate.

V. DRINA: NOVEL DATA ROUTING FOR IN-NETWORK AGGREGATION

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. In DRINA algorithm, the nodes sensing same event are formed as clusters. The node with highest energy among the other nodes in the cluster and with minimum distance to the sink node are chosen as cluster heads which acts as major role in aggregating the data and transmitting it to the sink node.

The proposed algorithm considers the following roles in the routing infrastructure creation:

1) Collaborator: A node that detects an event and reports the gathered data to a coordinator node.
2) 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.
3) Sink: A node interested in receiving data from a set of coordinator and collaborator nodes.
4) Relay: A node that forwards data toward the sink.

A. *DRINA is divided into three phases*

1) Updating the Hop count.
2) Formation of cluster and electing the coordinator.
3) Routing formation and node repair method.

VI. PHASE 1: NETWORK INITIALIZING AND UPDATING THE HOP COUNT

In this Module, 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. In this algorithm each sensor node have an id and hop to tree value by default with it which compares its own value with that of values contained in HCM message sent by sink node to other sensor nodes in a cluster.

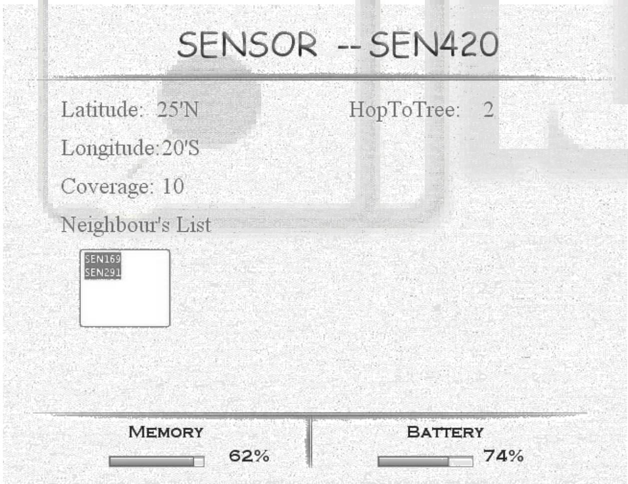

Fig. 1: Network Initializing and Updating the Hop Count

A. *Algorithm 1*

Hop Tree Configuration Phase

1) Node sink sends a broadcast of HCM messages with
   The value of HopToTree = 1.
   For each u ∈ R do
2) If HopToTree (u) > HopToTree (HCM) *and*
   First Sending (u) then
   Next Hop ← IDHCM;
   HopToTree ← HopToTreeHCM + 1;
3) Node u updates the value of the ID field in the message HCM.
   IDHCM ← IDu;
4) Node u updates the value of the HopToTree field in the Message HCM
   HopToTreeHCM ← HopToTree;
5) Node u sends a broadcast message of the HCM with
   The new values;
   FirstSendingu ← false;
6) End
7) Else
   Node u discards the received message HCM;
8) End
9) End

VII. PHASE 2: GROUPING OF CLUSTER AND ELECTING THE CLUSTER HEAD

Once the sensor node detects an event the event election algorithm starts. ALL the sensing nodes are eligible for the leader election algorithm .For the first event the leader is closest to the sink node or the leader is the node that is closest to the already established route. If tie occurs, the leader is the one with smallest id, if again tie occurs then the leader is the one with highest energy. Only one node will be detected as the leader the other nodes are the collaborators .Leader is the coordinator. The leader collects the data from the collaborator and sends them to the sink. The coordinator aggregates the information collected by the node sensing the same event. This is more efficient than the other aggregation mechanisms.

A. *Algorithm 2*

Grouping of cluster and electing the head.

1) Input: S
   S - Set of nodes that detected the event.
2) Output: u
   A node of the set S is elected leader of the group.
3) For each u ∈ S do
   Role u ← coordinator;
   Node u *sends message MCC in broadcast*
   Announcement of event detection;
   Nu *is the set of neighbors of node* u ∈ S
4) Node u retransmits the MCC message received from
   Node w;
   For each w ∈ N u do
   If HopToTree (u) > HopToTree (w) then
   Role u ← collaborator;
5) End
6) Node u retransmits the MCC message received from
   Node w;
   Else if HopToTree (u) = HopToTree (w) ^
   ID (u) > ID (w) then
   Role u ← collaborator;
7) End
8) Else
   Node u discards the MCC message received from w;
9) End
10) End
11) End
12) End

B. *Privacy preserving mechanism*

In existing work, encryption key was given only to the coordinator, so by convincing the cluster head the data can be collected by opponents [10] .But in our proposed system encryption key was given to each and every sensor nodes. Convincing every sensor node to get the data is not possible, thus, data are highly confidential.

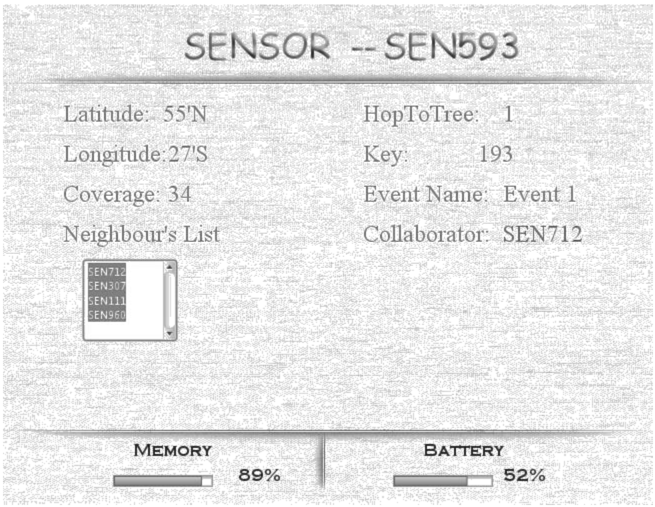

Fig. 2: Grouping Of Cluster and Electing the Cluster Head

VIII. PHASE 3: ROUTE FORMATION AND HOP TREE UPDATES

The leader elected will form the new route for event propagation. For the event propagation, the Coordinator sends a route establishment message to its NextHop node. The NextHop node retransmits the message to its NextHop and starts the hop tree updating process after when it receives a route establishment message. These steps are repeated until either the sink node is reached or a node that is part of a previously recognized route is found. The routes are formed by choosing the finest neighbor at each hop. The choice to choose the best neighbor are twofold: 1) The node that leads to the shortest path to the sink is chosen after the first event occurs and 2), the best neighbor is the one that leads to the closest node that is already part of an recognized route ,after the occurrence of subsequent events . The process tends to increase the aggregation points, ensures that the aggregation occurs as close as possible to the events. The ensuing route is a tree that connects the leader to the sink. When the route is recognized, the hop tree updating phase will begin.

The main goal of this phase is to update the hop tree to all the nodes; updation is done by the new relay nodes that are the part of the already recognized route. These nodes send a hop configuration message by means of a controlled flooding for the hop updating. The total cost of this process wills the same as that of the flooding, that is each node will send only one packet. The algorithm for the hop updating is same as that of the hop tree building algorithm.

The data transmission in this DRINA algorithm uses aggregation techniques that are applied in three different contexts:

1) Cluster within the opportunistic aggregation; 2) leader within the aggregation; and 3) cluster outside the aggregation. In the cluster within the opportunistic aggregation the routes overlap inside the cluster, the aggregation is performed by the collaborator nodes. In addition the coordinator performs data aggregation and sends the results to the sink node in the leader within the aggregation. In the cluster outside the aggregation, aggregation is performed by the relay nodes when two or more events overlap along routing.

The data transmission process is described in the algorithm. While the node has data to transmit, it verifies whether it has more than one successor that relays its data. It waits for a period of time and collects all data received and sends the aggregated data to its NextHop. Else, it ahead the data to its NextHop. For every packet transmission with aggregated data, the Route Repair Mechanism is executed. A route repair mechanism is used to send the data in a consistent way. Sender nodes wait for some time to receive a packet delivery confirmation. A new destination node is selected and the message is retransmitted by that node after when the sender node does not receive the confirmation.

A. Algorithm 3

Routing formation and hop tree update

1) Leader node v of the new event sends a message REM to its NextHop v;
2) Repeat
3) u is the node that received the REM message that was Sent by node v.
   If u = NextHop v then
   HopToTree u ← 0;
   Node u *is part of the new route built*
   Role u ← Relay;
   Node u sends the message REM to its NextHop u;
   Node u broadcasts the message HCM with the value Of HopToTree = 1; Nodes that receive the HCM Message sent by node u.
4) End
5) Until Find out the sink node or a node belonging to The routing structure already established.
6) Repeat
7) sons u is the number of descendants of u
   If sons u > 1 then
8) Aggregates all data and sends it to the NextHop u;
   If Role u = Relay then
   Execute the mechanism
9) End
10) End
11) Else
    Send data to NextHop u;
12) If Role u = Relay then
    Execute the mechanism
13) End
14) End
15) Until The node has data to transmit/retransmit;

IX. NODE REPAIR MECHANISM

The route created to send the data toward the sink node is distinctive and well-organized since it increases the points of aggregation and, thus, the information fusion. The route is distinctive, any failure in one of its nodes will cause trouble, prevent the delivery of several gathered event data. Potential causes of breakdown include physical demolition, low energy and communication obstruction. Some algorithms are based on periodic flooding mechanisms and fixed at the sink, to patch up broken paths and to find out new routes to forward traffic around faulty nodes. These mechanisms are not pleasing in terms of energy saving because it wastes a lot of energy with repairing messages. In addition, during the network flooding period, these algorithms are unable to route data around abortive 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 and sets a timeout, and then waits for the retransmission of the data packet by its NextHop. This re-transmission is a considered and an acknowledgement (ACK message). If the sender receives its acknowledgement from the NextHop node, it can conclude that the NextHop node is alive and, for now, the whole thing is ok. On the other hand if the sender node does not receive the ACK from the NextHop node within the determined timeout, it considers this node as dead 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, if tie occurs it chooses the neighbor with the maximum energy level. After that, the sender updates its routing table to smooth the progress of the forwarding of successive packets. A newly partial reconstructed path is created after the repairing mechanism is applied.

X. EXAMINING COMPLEXITY

Here, we derive the communication cost for DRINA, Information Fusion-based Role Assignment, and Shortest Path Tree algorithms. More particularly, we present the restrictions for the communication cost of these algorithms to create the routing structure. In addition we present the best and worst cases, and the average case will .In the Shortest Path Tree algorithms algorithm, there is constant communication cost to build the routing infrastructure. In an immediate way operation, it is essential one flooding started by the nodes that sensed the first event in order to build the routing tree. An extra flooding, initiated by the sink node, is also essential for the other nodes to set up their ancestors in the tree network. Thus, the standard communication cost for Shortest Path Tree is 2n, and n is number of nodes. The Information Fusion-based Role Assignment algorithm also introduces standard communication costs since it needs one flooding for every coordinator, preceded by one flooding initiated by the sink node. Each flooding is necessary to set up and update the aggregated cluster head-distance at each node. Moreover, it is essential m transmissions to create the group, where m is number of transmissions to create the group. For this reason, Information Fusion-based Role Assignment presents a constant communication cost of ð2kn þmÞ, here n represents the number of nodes, and k represents number of events. The overhead of the Information Fusion-based Role Assignment algorithm can be reduced in some situations by forcing a delay before each declaration of the coordinator is sent by the sink node. Thus, if consecutive events take place almost at the same time, only one flooding starting at the sink will be essential and the communication cost will be lower. In our designed , DRINA algorithm needs one flooding from the first elected coordinator and one more flooding initiated by the sink to create the initial tree infrastructure. Consecutive cluster-heads will make a scope-limited flooding to update the nodes HopToTree factor. Hence, the best case situation for the communication cost is when consecutive events take place near the formerly recognized routing tree. The best case will be achieved when the events happen on the routing tree. In this case, each CH is already attached to the tree. The number of transmissions to establish the initial routing tree is 2n plus m transmissions to create the cluster, i.e., the cost is ð2n þmÞ. The worst case of the DRINA algorithm happens when successive events take place far from the previously created tree. Here, the number of transmissions to build the initial tree is 2n in addition ðk _ 1Þn _Pki¼2 jUij transmissions for the following events in addition m messages to create the cluster, where n represents number of nodes, k is number of events, m represents number of transmissions to create the clusters and jUij is the cardinality of the set of nodes exterior to the scope-limited flooding for the event i, which will not update the HopToTree for that event. Hence, the worst case for the DRINA algorithm is ð2n þ ððk _ 1Þn _Pki¼2 jUijÞ þ mÞ.

XI. CONCLUSION

Data aggregation plays a significant role in wireless sensor networks. Our novel and reliable data aggregation aware routing protocol is compared with Shortest Path Tree algorithm, LEACH algorithm and Information Fusion-based Role Assignment algorithm. Our proposed algorithm has maximized the aggregation of packets and also delivering the information in case of node failure. It includes privacy preserving of information where every sensor node has a key apart from the key of the coordinator node.

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] 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. de/db/conf/icc/icc2009.html#AbdelSalamO09, 2009.
[4] L. Villas, A. Boukerche, R.B. de Araujo, and A.A.F. Loureiro, "Highly Dynamic Routing Protocol for Data Aggregation in Sensor Networks," Proc. IEEE Symp. Computers and Comm. (ISCC), pp. 496-502, http://dx.doi.org/10.1109/ISCC.2010.5546580, 2010.
[5] L.A. Villas, A. Boukerche, H.A. de Oliveira, R.B. de Araujo, and
[6] I. Chatzigiannakis, S. Nikoletseas, and P.G. Spirakis, "Efficient and Robust Protocols for Local Detection and Propagation in Smart Dust Networks," Mobile Networks and Applications, vol. 10,8nos. 1/2, pp. 133-149, 2005.
[7] C. Efthymiou, S. Nikoletseas, and J. Rolim, "Energy Balanced Data propagation in Wireless Sensor

Networks," Wireless Networks, vol.12, no. 6, pp. 691-707, 2006.

[8] L.A. Villas, D.L. Guidoni, R.B. Arau´ Jo, A. Boukerche, and A.A.Loureiro, "A Scalable and Dynamic Data Aggregation Aware Routing Protocol for Wireless Sensor Networks," Proc. 13th ACMInt'l Conf. Modeling, Analysis, and Simulation of Wireless and Mobile Systems,pp.110-117,http://doi.acm.org/10.1145/1868521. 1868540, 2010.

[9] E.F. Nakamura, A.A.F. Loureiro, and A.C. Frery, "Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications," ACM Computing Surveys, vol. 39, no. 3, pp. 9-1/9-55, 2007

[10] Leandro Villas, Azzedine Boukerche1, Heitor S.Ramos, "DRINA: A Lightweight and Reliable Routing Approach for in-Network Aggregation in Wireless Sensor Networks" *Wireless Sensors, 2013*.