Data Routing In-Network Aggregation for Wireless Sensor Network

G.M. Joshi
M.B.E.S College of Engineering, Ambajogai
Maharashtra, India

B.M. Patil
M.B.E.S College of Engineering, Ambajogai
Maharashtra, India

ABSTRACT

WSNs have limited computational power, limited memory and battery power this increases the complexity and leads to need for data aggregation method. The main goal of data aggregation algorithms is to gather and aggregate data in an energy efficient manner, so that lifetime of network is increased. When WSN sensing an event the redundant data will be detected and collected this need to increase in communication cost and energy consumption of network so, in this work the DRINA (a novel data routing for In-network Aggregation) protocol has some advantages like a reduced number of messages for setting up a routing tree, high aggregation rate, maximized number of overlapping routes. The DRINA algorithm was compared with two other algorithms: (InFRA) The Information Fusion-based Role Assignment and Shortest Path Tree (SPT) algorithms it provides best result.

Keywords

Wireless sensor networks, DRINA, InFRA, SPT.

1. INTRODUCTION

A Wireless sensor networks (WSNs) contains hundreds or thousands of sensor nodes. They are used to monitor the sensing field and collect information from physical or environmental condition, and pass the collected data through the network to a main location such as temperature, pressure, vibration, pollutants or motion at distinct places [1] [2]. WSNs have been used in various application areas like military, security, critical infrastructure system, communication and health [3][4].WSNs are also a special type of ad hoc networks. The wireless medium can be formed by infrared, radio or optical media. WSN are data driven networks and generate large information that needed to be forward generally in multihop pattern towards a sink node, where sink node performs a role of controlling station [5]. The data aggregation aware routing algorithms play an essential role in WSN. Data aggregation is an effective technique for saving energy in WSNs, for more efficient and effective data gathering with a minimum use of resources, sensor nodes should be configured to smartly report data by making local decision[6][7].

In-network data aggregation is main technique supported by wireless sensor networks. In-network data aggregation is the global process of gathering and routing information through a multi-hop network, processing data at the intermediate nodes by reducing resource consumption and increasinging the network lifetime. In-network data aggregation can be used due to the redundancy in raw data gathering, decrease the communication cost and forward only smaller aggregated data towards sink node since it leads directly to energy saving and extend the network lifetime. The main use of data aggregation is to reduce the data redundancy, communication cost and increase data accuracy and save energy [8].

The main challenge of WSN is how to guarantee the data delivery even node failures and interruption in communication. These failures become more critical when aggregation is performed. Whenever one of these packets is lost a considerable amount of information will also be lost. Realization of sensor network needs to satisfy the constraints introduced by the factors such as fault tolerance, to overcome these challenges the DRINA algorithm is used for WSNs. This paper is organized as follows in section 2.literature Review. Section 3 introduction of DRINA algorithm .In section 4 simulation and performance analysis scenarios section 5 conclusions and future work.

2. LITERATURE REVIEV

Three main timing strategies are found in the literature given below [9][10].

Periodic simple aggregation: Each node waits for a predefined period of time while aggregating all received data packet and then forward a single packet with the result of the aggregation.

Periodic per-hop aggregation: This approach requires each node to know the number of its children's. It is quite similar to the periodic simple aggregation, but the aggregated data packets transmitted as soon as the node hears from all of its children. In this case some children's packet being lost.

Periodic per-hop adjusted aggregation: In this strategy adjusts the transmission time of a node according to this node's position in the gathering tree.

Various approaches have been proposed to provide data aggregation during the routing in WSNs. Some of them are tree-based approach, cluster-based approach and structure-less approach.

2.1 Tree-Based Approaches

Tree-based protocols build a traditional shortest path routing tree, it uses the Shortest Path Tree (SPT) algorithm [11]. This is a very simple strategy to build a routing tree in a distributed fashion. In Shortest path tree (SPT) data aggregation scheme, each source sends its information to the sink along the shortest path. This approach has some drawbacks, For instance, when a packet is lost at any level of the tree, data from the whole sub tree will be lost because of this tree-based approaches require a fault tolerance mechanism to reliably forward the aggregated data the Greedy Incremental Tree (GIT) [12] approach was proposed.

The Greedy Incremental Tree (GIT) algorithm was proposed that based on directed diffusion. The GIT algorithm establishes an energy-efficient path and greedily attaches other sources onto the established path. In the GIT strategy, nodes send their information as in the SPT algorithm, when the first event is detected and, for every new event, the information is routed using the shortest path to the current tree.In the TAG algorithm, parents notify their children about the waiting time for gathering all the data before transmission. Data aggregation is implemented in a real-world testbed and the Tiny Aggregation Service (TAG) approach is established [13] [14].

2.2 Cluster-Based Approaches

Cluster-based schemes [15] also consist of a hierarchical organization of the network. In these approaches, nodes are divided into clusters. Special nodes, referred to as cluster-heads, which are elected to aggregate data locally and forward the aggregation result to the sink node. In the Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm [16]. In this algorithm, cluster heads can act as aggregation points and they communicate directly to the sink node. LEACH-based algorithm limits the size of the network.

The Information Fusion-based Role Assignment (InFRA) algorithm builds the shortest path tree that maximizes information fusion. When multiple nodes detect the same event, they organize themselves into clusters. A disadvantage of the InFRA algorithm is 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 for each new event that arises in the network DRINA algorithm is also a cluster – based approach, but it works differently from the InFRA algorithm, that it does not flood a message to the whole network whenever a new event occurs. DRINA builds a routing tree with the shortest paths that connect all source nodes to the sink while maximizing data aggregation.

2.3 Structure-Less Approaches

A structure-less approach uses the Data-Aware Anycast (DAA) algorithm [17], uses anycast to forward packets to one-hop neighbors that have packets for aggregation. This approach does not guarantee aggregation of all packets, the cost of transmitting packets with no aggregation increases in larger networks.

3. DRINA: DATA ROUTING FOR IN-NETWORK AGGREGATION FOR WSNS

The DRINA algorithm is a cluster based technique. It exploits data aggregation technique in which all source nodes connects with the sink via shortest path. When source node has to send data to the sink node, they form a cluster. One cluster head is elected from cluster. It aggregates all the data from remaining nodes in cluster and forms path up to sink node via shortest path Routing tree cost and communication cost for DRINA is less. Different nodes have different role in the network.

1) Coordinator: - These nodes collect all data packets from collaborator nodes, aggregate them and forward towards sink node.
2) Collaborator:-These nodes send data packets to the coordinator node.
3) Relay:-These are middle nodes between coordinator and sink.
4) Sink:-It is gateway to monitoring centre and interested in data collected from coordinators and Collaborators.

This algorithm constructs the routing tree in terms of hops. It involves three phases. The following Fig (1) shows the architecture of DRINA.

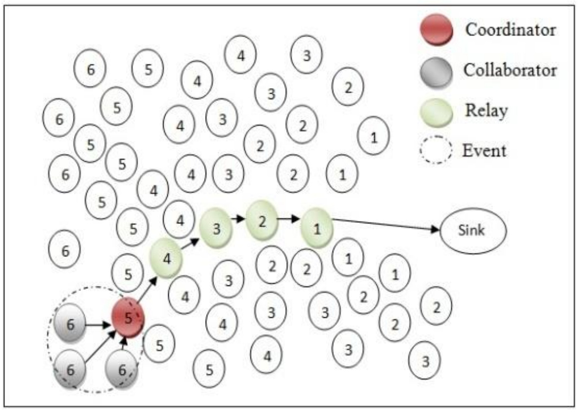


Fig (1): DRINA Architecture

3.1 Hop Tree Construction phase

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. This HCM consists of two fields: HopToTree Value and Node Identifier. HopToTree value is a value which is a distance from the sink. It constructs the route depending on the minimum HopToTree value from the sink. If already constructed, route has minimum HopToTree value than the present HopToTree value then the route never changes. Previous route must be preferable.

Algorithm 1: HOP tree Configuration Phase

```
Node sink send a broadcast of HCM messages with value of HopToTree = 1;
for each $u \epsilon R_u$ do
    if HopToTree (u) > HopToTree (HCM) and First Sending (u) then
        $NextHop_u \leftarrow ID_{HCM}$ ;
        $HopToTree_u \leftarrow HopToTree_{HCM} + 1$;
        $ID_{HCM} \leftarrow ID_u$;
        $HopToTree_{HCM} \leftarrow HopToTree_u$;
        Node u sends a broadcast message of the HCM with the new values;
        First $Sending_u \leftarrow$ false;
    End
    Else
        Node u discards the received massage HCM;
    End
end
```

3.2 Cluster Formation

After the hop tree construction, cluster election algorithm should be made. The main thing in this cluster formation is to select the cluster head. Based on the following criteria: 1. Cluster head is a node which is immediate to the sink. 2. If one node is immediate to the sink, then the tie arises. Basically each node contains the ID (identifier).The node which has least possible ID will revolve as a cluster head. 3. If tie arises, precedent constructed shortest route as a cluster head. This node is in turn dispatched to the neighbor Next Hop node. This process is reciprocated until all the nodes in the network have reached. After routes are established, HopTree is updated. Relay nodes are bonded for this process.

Algorithm 2: Cluster formation and leader election

```
Input: S
Output: u
for each $u \in S$ do
  $role_u \leftarrow$ coordinator;
  Announcement of event detection;
  foreach $w \in N_u$ do
    if HopToTree $(u)$ >HopToTree$(w)$ then
      roleu ← collaborator;
      Node u retransmits the MCC message received from node w
    End
    else if HopToTree $(u)$ = HopToTree$(w)$ $\wedge$ ID(u) > ID(w) then
      $role_u \leftarrow$ collaborator ;
      Node u retransmits the MCC message received from node w;
    End
    else
      Node u discards the MCC message received from $w$;
    End
  End
end
```

3.3 Routing Formation and Hop Tree Updates

In this phase the elected group leader i.e. coordinator starts establishing the new route, that means 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 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.
2) Choose the best neighbor is the one that is already part of an established route.

This process helps to increase in aggregation data. 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, described in Section 3.

Algorithm 3. Route establishment and hop tree update

```
Leader node $v$ of the new event sends a message REM to its NextHop
Repeat
  if $u$ = NextHop$_v$ then
    $HopToTree_u \leftarrow 0$ ;
    $Role_u \leftarrow Relay$ ;
    Node $u$ sends the message REM to its $NextHop_u$;
    Node $u$ broadcast the message HCM with the value of HopToTree = 1;
    Nodes that receive the HCM message send by node u, will run the command line 2 until the 14 of Algorithm 1;
  End
until find out the sink node or a node belonging to the routing structure already established;
Repeat
  if $sons_u > 1$ then
    Aggregates all data and sends it to the $nexthop_u$ ;
    if $Role_u = Relay$ then
      Execute the route repair mechanism
    End
  End
  Else
    send data to nexthop$_u$ ;
    if $Role_u = Relay$ then
      Execute the route repair mechanism
    End
  End
until the node has data to transmit/retransmit;
```

3.4 Route Repair Mechanism

The Route Repair Mechanism means sender nodes waits for a predefined time period to receive a packet delivery conformation, when the conformation is not received by the sender node a new destination node is selected and the message is retransmitted through that node. 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 [18] [19], 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.

DRINA algorithm offers a piggybacked, ACK-based, route repair mechanism, which consists of two parts: failure detection at the Next Hop node, and selection of a new Next Hop 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. As an example, a disrupted route is shown in Fig (2). After the repairing mechanism is applied, a newly partial reconstructed path is created as depicted in Fig (3).

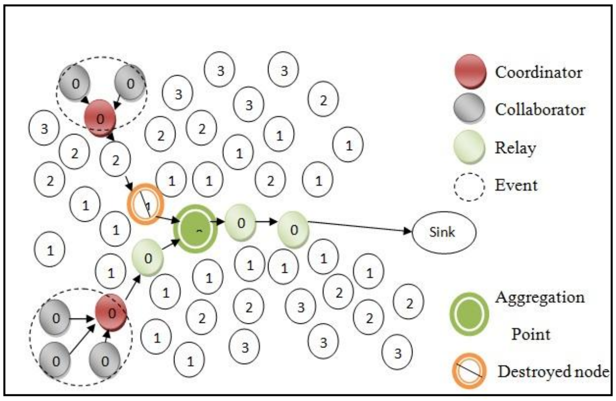


Fig (2): Region with destroyed nodes

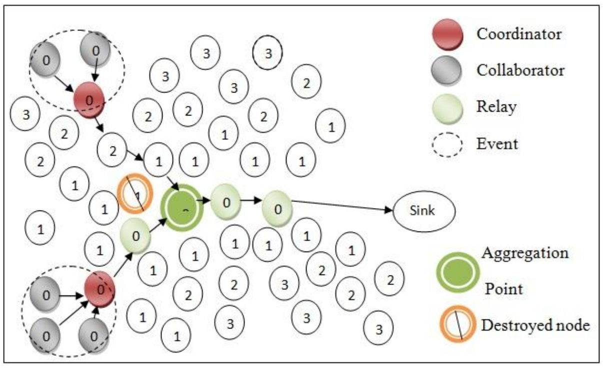


Fig (3): Repaired path

4. SIMULATION AND PERFORMANCE ANALYSIS

4.1 Simulation

The simulation scenario of DRINA algorithm shows the following results. Fig (4) shows simulation window having thirty nodes and it work for assign node to elect cluster head. Firstly the sink node sends HCM (Hop Configuration Message) to whole network by flooding method. Once election of cluster head is completed the cluster formation phase will be start by choosing the coordinator, collaborator and relay node. The leader node is one that is closest to the sink node. The leader is coordinators that collects the data from different nodes and send that collected data to the sink node, which is shown in Fig (5).

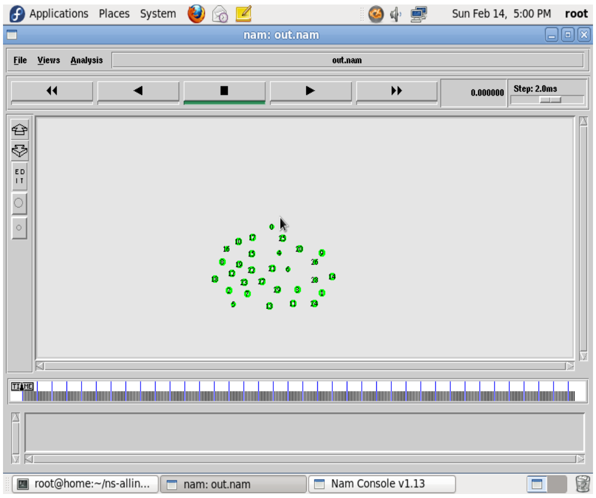


Fig (4): Assign node to elect cluster head

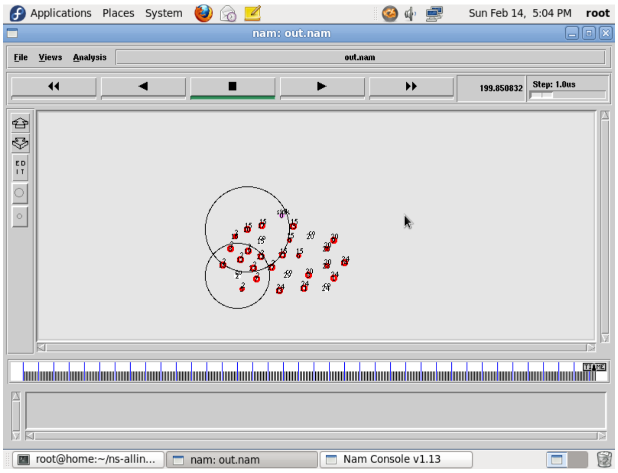


Fig (5): Formation of cluster and cluster head selection

4.2 Performance Analysis

In this section, the DRINA algorithm compares its performance to other two known routing algorithms: the InFRA and SPT algorithms. The performance of DRINA algorithm evaluate under the following metrics:

1. Packet delivery rate.

2. Control overhead.

3. Routing tree cost.

4.2.1. Packet delivery rate:

PDR (Packer Delivery Rate) is expressed as a ratio of Total number of packets sent by all sources to the total number packets received by the sink. PDR indicates the quality of the routing tree built by algorithms. DRINA has more PDR than SPT and InFRA algorithms. The fig (6) shows the performance of these three algorithms. The nature of graph is number of nodes verses PDR shows that as the number of nodes increases the PDR also increases.

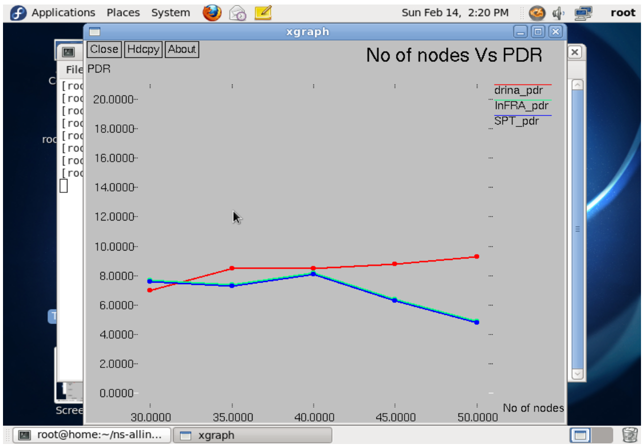
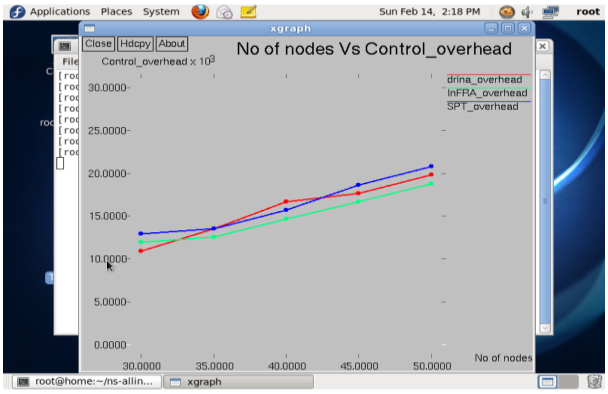


Fig (6): Graph with Number of Nodes Vs PDR

4.2.2. Control overhead:

CO (Control Overhead) is expressed as the ratio of total number of control packets forwarded by the sender to the number of data packets delivered to the receivers. The following graph shows the performance of SPT, InFRA and DRINA algorithms. The nature of graph is number of nodes verses control overhead shows that as the number of nodes increases control overhead of DRINA decreased. The nature of graph is shown in Fig (7).



.Fig (7): Graph With Number of Nodes Vs Control_overhead

4.2.3. Routing Tree Cost

The routing tree cost is calculated by the number of edges constructed by the algorithm. . The nature of graph is number of nodes verses routing tree cost shows that as the number of nodes increases tree cost of DRINA increases. The nature of graph is shown in Fig (8). The routing tree performance of SPT, InFRA and DRINA algorithms shown below.

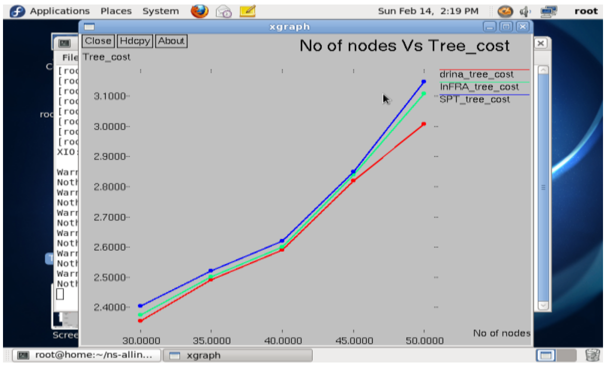


Fig (8): Graph With Number of Nodes Vs Tree_cost

5. CONCLUSION

Data Routing for In-Network Aggregation (DRINA) is efficient and reliable data aggregation aware routing protocol for WSNs. This algorithm is compared with InFRA and SPT algorithms. The comparison parameters are scalability, communication costs, delivery efficiency, aggregation rate, and aggregated data delivery rate. The comparison clearly shows that total number of packets transmitted in DRINA is very much less than InFRA and SPT. So, the energy consumption for DRINA algorithm is less than other two compared algorithms .The obtained Packet delivery ratio shows that DRINA sends data in reliable way as it use Route Repair Mechanism technique. Small amount of packets are lost during simulation.

Thus an overall simulation result concludes that DRINA can increase network life as it required low energy and tree cost. Also, show that the proposed algorithm has some key aspects required by WSNs algorithms such as maximized number of overlapping routes, high aggregation rate, Reliable data aggregation and transmission.

6. 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] L. A. Villas, A. Boukerche, H. S. Ramos, Horacio A.B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, “DRINA: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks,” IEEE Trans. on Computers, vol. 62, no. 4, pp. 676-689, April 2013.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, pp. 393-422, 2002.

[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] I. Chatzigiannakis, T. Dimitriou, S.E. Nikoletseas, and P.G. Spirakis, “A Probabilistic Algorithm for Efficient and Robust Data Propagation in Wireless Sensor Networks,” Ad Hoc Networks, vol. 4, no. 5, pp. 621-635, 2006.

[7] 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, Nos.1/2, pp. 133-149, 2005.

[8] 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.

[9] F. Hu, X. Cao, and C. May, “Optimized Scheduling for Data Aggregation in Wireless Sensor Networks,” Proc. Int’l Conf. Information Technology: Coding and Computing (ITCC ’05), pp. 557-561, 2005.

[10] I. Solis and K. Obraczka, “The Impact of Timing in Data Aggregation for Sensor Networks,” IEEE Int’l Conf. Comm., vol. 6, pp. 3640-3645, June 2004.

[11] B. Krishnamachari, D. Estrin, and S.B. Wicker, “The Impact of Data Aggregation in Wireless Sensor Networks,” Proc. 22nd Int’l Conf. Distributed Computing Systems (ICDCSW ’02), pp. 575-578, 2002.

[12] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, “Impact of Network Density on Data Aggregation in Wireless Sensor Networks,” Proc. 22nd Int’l Conf. Distributed Computing Systems, pp. 457-458, 2002.

[13] E.F. Nakamura, H.A.B.F. de Oliveira, L.F. Pontello, and A.A.F. Loureiro, “On Demand Role Assignment for Event-Detection in Sensor Networks,” Proc. IEEE 11th Symp. Computers and Comm. (ISCC ’06), pp. 941-947, 2006.

[14] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, “Tag: A Tiny Aggregation Service for Ad-Hoc Sensor Networks,” ACM SIGOPS Operating Systems Rev., vol. 36, no. SI, pp. 131-146, 2002.

[15] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler, “Supporting Aggregate Queries over Ad-Hoc Wireless Sensor Networks,” Proc. IEEE Fourth Workshop Mobile

Computing Systems and Applications (WMCSA ’02), pp. 49-58, 2002.

[16] A.P. Chandrakasan, A.C. Smith, and W.B. Heinzelman, “An Application-Specific Protocol Architecture for Wireless Micro sensor Networks,” IEEE Trans. Wireless Comm., vol. 1, no. 4,pp. 660-670, Oct. 2002.

[17] K.-W. Fan, S. Liu, and P. Sinha, “On the Potential Of Structure-Free Data Aggregation in Sensor Networks,” Proc. IEEE INFOCOM,pp. 1-12, Apr. 2006.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” Proc. MobiCom, pp. 56-67, 2000.

[19] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System Architecture Directions for Networked Sensors,” ACM SIGPLAN Notices, vol. 35, no. 11, pp. 93-104, 2000.