FEDA: Fault-tolerant Energy-Efficient Data Aggregation in Wireless Sensor Networks

Mohammad Hossein Anisi
Department of Electrical and Computer Engineering
Azad University of Qazvin, Qazvin, Iran
anisi@qazviniau.ac.ir

Javad Rezazadeh
Department of Electrical and Computer Engineering
Azad University of Qazvin, Qazvin, Iran
rezazadeh@qazviniau.ac.ir

Mehdi Dehghan
Department of IT and Computer Engineering
Amirkabir University of Technology, Tehran, Iran
dehghan@aut.ac.ir

Abstract— **one of the key issues in wireless sensor networks is data collection from sensors .In this area, data aggregation is an important technique for reducing the energy consumption. Also, reliability and robustness of transferring data is one of the important challenges. The in-network data aggregation approach which is proposed in this paper, besides achieving ideal energy consumption by limiting a number of redundant and unnecessary responses from the sensor nodes, it can increase the chance of receiving data packets at the destination and cause to a more accurate results. By utilization of J-Sim simulator, the proposed approach is compared and evaluated with some important approaches in this area. The simulation results show that by using the proposed approach, the loss amount of the data packets and the average energy consumption of network will be considerably reduced.**

I. INTRODUCTION

Wireless Sensor Networks consist of several sensor enable nodes which are distributed in an environment and use batteries as energy resource. In such Networks, you can query about physical quantities of the environment. The sensors nodes can be in a tree liked structure that with in, the base station roles as the root of the tree and each node has a parent. Therefore, the data items can be transmitted hop by hop from the leaf nodes to the root and also they can aggregate during such transmission [7], [16].

Because of using common and wireless Medias and appearing of some environment causes like noises, data transmission between the network nodes always expose to faults and loss of packets. The loss of data has a direct effect on the accuracy of the received result, because the part of the values which should use in partial data aggregation became loss. Especially when such data losses occurs near the root the amount of lost data become larger and the result become more inaccurate.

In our approach, when a packet is lost between two sensors because of a link error, it is possible that one or more other sensors have correctly overheard the packet. If some of them are yet to send their own values, they correct the error by aggregating the missing value into theirs. As a result, error recovery has no overhead because the lost packet is aggregated with another packet to be transmitted

The proposed approach which is called FEDA (Fault-tolerant Energy-efficient Data Aggregation), by considering the mentioned challenges, reduce the impact of them considerably. In this robust approach, by limiting a number of redundant and unnecessary responses from the sensor nodes, the energy-efficiency as well as provides an accurate response will be significantly improved.

In continuation, in section 2, the related works of data collection in sensor networks especially those which are fault tolerant will be presented. In section 3 the model of the system and in section 4 the proposed approach in the name of FEDA will be presented. In section 5, our approach will be evaluated and also the simulation results will be presented.

II. RELATED WORKS

There are several approaches which use tree structure for collecting and aggregating data. The presented approach in [5], with combining Clustering and Directed Diffusion Protocol [6], could process, collect and aggregate the data of sensor node without any dependency to the related environment. This paper, with presenting a dynamic clustering structure, could enable the nodes to join the nearest head cluster when they want to send data to the gateway node.

In the TAG (Tiny Aggregation) approach [7], each epoch divides to some time slots and these time slots specify to different levels of routing tree in reversal form. In this manner, each node depends on its situation in the tree, and in its related time slot will send its data. The node synchronization of this approach for sending and receiving data could effectively reduce the average energy consumption. The big problem of this approach is that if the information of a child node can not be received by the parent node, the whole information of the sub tree, which the child node was the root of it, will be lost and finally a wrong or inaccurate result will be received.

In the BDA (Bidirectional Data Aggregation) [8], approach which uses the basic of TAG, they add a Label to each query that lead to omitting some additional transmissions and getting better energy consumption. We also take advantage of this Label in our approach. But, this approach besides having good energy consumption has the same problem with the TAG approach that has no strategy for preventing faults and inaccurate results.

There are also several approaches about reliable data collection in sensor networks [2], [4], [9-14]. In [10] they reduce the impact of transient faults on the links and also the loss of packets by retransmission of loss packets but this approach has a large overhead either in time and energy. There are also other approaches that could make reliable receiving of data almost possible by using some techniques such as retransmission [12], [13] and multiple paths [2], [4], [14]. The problem of such approaches is that the response time of them is high and also because the lack of node synchronization in the different levels of the tree, the average consumption is not appropriate.

III. FAULT-TOLERANT ENERGY EFFICIENT DATA AGGREGATION (FEDA)

One of the issues that need energy consumption is data collection form sensor nodes. In this case, there is the probability to receive some same data or some redundant data multiple times. For example, when you want to find the max value of sensors, you need only one value that the other values which are the same or smaller than the max value are redundant. However, sending such redundant messages will increase the energy consumption of network. The proposed approach, omit the redundant responses and as a result reduce the energy consumption of network.

On the other hand, in tree-based approaches, when a response packet is lost because of a link error, a complete sub tree of values is lost, possibly leading to an incorrect aggregate result [2], [3], [7]. Our proposed approach uses the available path redundancy in the wireless sensor networks to deliver a correct aggregate result to the data sink.

We study our proposed approach in three distinct parts. In section the mechanism of building aggregation tree will be studied. In section B, the error recovery mechanism of approach and in section C, the whole operation of approach will be studied.

A. Building Aggeragation Tree

The process of building aggregation tree of our proposed approach is like other similar approaches with a little difference. This phase starts with sending a build tree message from the gateway node. The gateway node considers its level 0 broadcasts the message. The nodes which receive this message select the gateway node as their father and select their level one more than the level of their father. During this process, each node selects parent and a routing tree will be formed that can use for returning the responses of nodes to the root. But, in our approach there is a difference that if a node receives a build message twice and if it selected its father before, instead of throwing the message away, it selected the sender of the second message as its back up parent and drop all other received messages. However, in the return phase, each child node sends its responses to both of its parents. Algorithm1 simply shows the process of building such aggregation tree.

1. The gateway node broadcast the *build message* as the first node.
2. **For each *build message* receiver:**
   - **a. If the node receives the message for the first time:** The sender node will be selected as the primary parent, *Node_Level = Parent_Level +1,* broadcast the *build message and wait for D.T for receiving the responses of child nodes.*
   - **b. Else if the node receives the message for the second time:** it chooses the second sender as its back up parent.
   - **c. Else,** drop all the received messages**.**

Algorithm 1. The phase of building aggregation tree

B. Error Recovery

In our approach, edges are classified into three types: primary, backup, and side edges; primary and backup edges are between a sensor node and its parents and side edges are among parents. Each sensor selects one parent as its primary parent and zero or one parent as backup. If an error occurs in a primary edge, it is possible that some backup edges have successfully delivered the sent value. Parents coordinate using side edges so that the missing value is aggregated at most once (i.e., no more than 100% of the value is aggregated). It should be noted that a sensor can be a primary parent for some children and at the same time a backup parent for some others. Also, we assume that errors occur independently in primary, backup and/or side edges.

Also, in the proposed approach, each packet contains one additional field that related to the successful or unsuccessful receive of primary parent of each child node which is called FS (Fault status). When a primary parent does not receive a packet from its child, it will set the FS field of its own message to 1. The adjacent node of primary parent, which can be the back up parent of the child node, can detect this error by snooping on the side edge which is between them. However, it will aggregate that value to its local value before sending its own messages. Algorithm 2 shows this process.

1. **For each node as a primary parent:**

   a. **If it received a packet from a child**:
   FS=0;
   Sends the aggregated value of its own local value and the value of its child toward the root.

   b. **Else:**
   FS=1;
   Sends its own local value toward the root.

2. **For each node as a Back up parent:**

   c. **If it found the FS field of its sibling node equal to zero:**;
   Sends its own local value toward the root and Drop the back up value.

   d. **Else:**
   Sends the aggregated value of its own local value and the back up value

Algorithm 2. Returning Results

C. In-Network Data Aggregation

Consider a network of tree rooted from a seek node to leaf nodes. Our proposed algorithm can function in any topology of tree. We assumed that the aggregation tree is formed at the network initialization phase. Also we consider the depth of tree D, the maximum allowable round-trip time between two neighboring nodes T and d as the depth of the node which is the number of edges from the sink to it.

When the gateway disseminates a query for the child nodes, it will wait for D.T for receiving the responses of child nodes. Then, after each node receive the query; it will compare the Label of the query with the value of its sensor. If the Label value was redundant, it will update the Label with its own value and forward the query toward its children; otherwise, it will forward it without any update. Also, it will wait for (D-d).T for receiving the response. When it didn't receive the response up to this time, it will send it own value to the gateway.

When the query reaches at a leaf node, the leaf node compares its local value with the Label. If the Label value was redundant, it will update the Label with its own value and forward the query toward its parents (Primary and back up parents); otherwise, it will send no response. Also, when an intermediate node receives a response from its children, if the Label value becomes redundant, it will update the Label with its own value and forward the query toward its parents. But, if it doesn't receive any response up to the specified time ((D-t).T), it will send it own local value to its parents. For realizing the approach better, we can study an example that gathers the MAX aggregate:

In figure 1, the transmissions and some of back up transmissions are shown. Consider node B and C which are sibling nodes. They compare their local value with the value of query label and as their local values are greater than label, they update the label with their own local values and node B forwards 36 and node C forwards 32 to their children. As it is shown in the figure, they also send their values to their stepchild. Also, if you consider nodes D and E, they do the same process again.

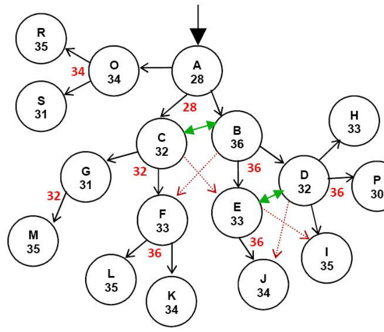

Figure 1. Querying the MAX aggregate value. In the figure, the slotted arrows show the back up transmissions, the bidirectional arrows show the communications between sibling nodes, the values in the circle show the local values and the values on the arrows show the value of Label.

In figure 2, node M was the only node which has a value greater than the received Label; however, it will send up its local value toward the root. In continue, other nodes which have not received any packets up to their specified time, send their own local value toward the root, for example, node D and E which updated their own local value with 36, send 36 to both of their parents.

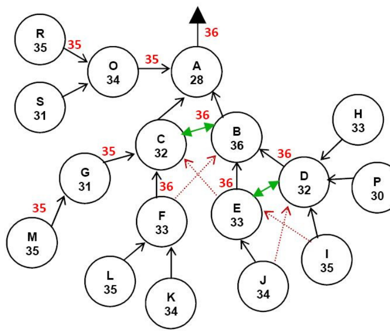

Figure 2. Returning the results toward the root.

In this approach, the Snooping technique [16] can be also used that allows nodes to locally suppress local aggregate values by listening to the answers that Neighboring nodes report and exploiting the semantics of aggregate functions. For example, in figure 7, node C can score its `MAX` value low when it hears a `MAX` from node 3 that is larger than its own. For dense network topologies where there is ample opportunity for snooping, this technique produces a dramatic reduction in communication, since at every intermediate point in the routing tree, only a small number of node's values will actually need to be transmitted.

Also, as the nodes expose to dynamic sensor readings in the real environment, if a node received better values during its waiting time, it can update its local value with the best one before receiving the values of its children and sending its response.

IV. PERFORMANCE EVALUATION

The proposed approach is simulated and evaluated with J-Sim (Java-Based simulator) [15]. We used the TAG approach as a famous In-Network data aggregation approach and BDA approach which is also an energy-efficient In-Network data aggregation approach, which have not any fault tolerant mechanism, and also, a tree-based data aggregation approach that uses retransmission TAG-Re (TAG with Retransmission) as a robust approach for evaluating the proposed approaches.

Two factors of network size and the fault ratio affect the number of lost packets. By doing several simulations and experiments and by changing two mentioned factors we checked the results. First, we considered the fault ratio two constant values of 0.5% and 1.5% and change the network size with the size of 16, 64, 256, 512 and 1024 nodes. Then, we considered the network size the constant value of 256 nodes and changed the fault ratio from 0.4% to 1.5%. We evaluated the four mentioned approaches by Root Mean Square (RMS) and the average energy consumption benchmarks. The simulation results presents that the FEDA approach can highly improve the accuracy of results and also the average energy consumption.

We used the formula in [18] for calculating the energy consumption in sending each packet as below:

$Energy = m \times size + b$

That within, size is the packet size, m is the required energy for sending each bit and b is the required energy to prepare a packet for sending. In the simulations, for all the approaches, we considered m as 10 and b equal to 100 nano joules.

Figures 3 and 4 present the average energy consumption in networks with different network size and in two different fault ratios, one with a lower fault ration of 0.5% and another with higher fault ration of 1.5%. As what as observed in the figures, when the fault ratio is low (figure 3), the average energy consumption of FEDA is lower than other approaches that the reason is omitting some redundant responses which is explained in part B of section 3. Also, in the TAG with retransmission, as the number of successful transmission is high and the packets arrive at destination successfully, there is no need for retransmissions; however, the average Energy consumption is approximately like the basic TAG.

But when the fault ratio goes higher (Figures 4) and the number of unsuccessful transmission become higher, the ability of FEDA approach becomes more obvious. As it can see in the figure, in the TAG with Retransmission, because of using retransmission and increasing the number of them, the more energy should consume. In the FEDA approach, there is no retransmission; however, the average energy consumption is still the same as before and lowers than the other approaches.

Figure 5 presents the amount of Root Mean Square error in different approaches with different fault ratios in a network with the size of 256 nodes. As the figure presents, the FEDA approach have the minimum RMS error which is almost insignificant that proves the robustness of these approaches in higher fault ratios.

So, it can be concluded that the FEDA approach, with owning the minimum RMS error and also owning the lowest average energy consumption, comparing to other presented approaches, is the best.

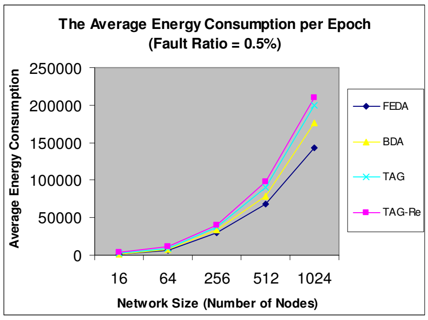

Figure 3. The average energy consumption per epoch with fault ratio of 0.5%

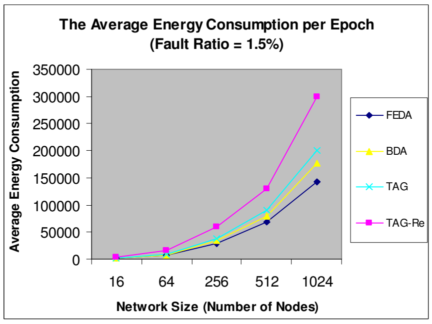

Figure 4. The average energy consumption per epoch with fault ratio of 1.5%

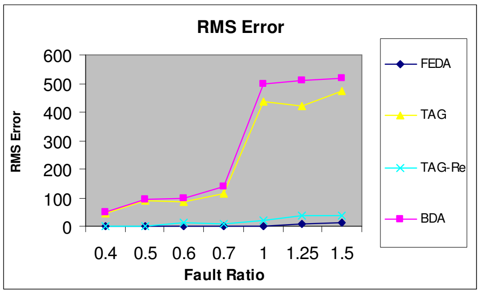

Figure 5. The RMS error with different fault ratios

V. CONCLUSION

In the proposed approach which is presented in this paper, we could limit a number of redundant and unnecessary responses from the sensor nodes. Also this scheme makes use of the available path redundancy in the network to deliver a correct aggregate result to the data sink. According to the simulation results, we proved that the proposed approach with having an appropriate Energy consumption and also minimum RMS error can be an effective approach in wireless sensor networks

REFERENCES

[1] Young Yao, Johannes Gehrke, "Query Processing for Sensor Networks," Proceedings of the First Biennial Conference on Innovative Data Systems Research, Asilomar (CIDR '03), Ca, USA, January 2003.

[2] J. Considine, F. Li , G. Kollios , J. Byers, "Approximate Aggregation Techniques for Sensor Databases," Proceedings of the 20th International Conference on Data Engineering, Boston University, MA, USA, April 2004.

[3] A. Manjhi, S. Nath, P.B. Gibbons, "Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams," Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 2005.

[4] S. Nath, P.B. Gibbons, S. Seshan, Z.R. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," Proceedings of the 2nd Int. Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, November 2004.

[5] S. Chatterjea, P. Havinga, "A Dynamic Data Aggregation Scheme for Wireless Sensor Networks," Proc. Program for Research on Integrated Systems and Circuits, (ProRISC 2003), Veldhoven, Netherlands, November 2003.

[6] Ch. Intanagonwiwat, R. Govindan , D. Estrin, "Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks," Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, Massachusetts, USA, August 2000.

[7] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks," Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Boston, Massachusetts, USA, December 2002.

[8] S. Cho, "Bidirectional data aggregation scheme for wireless sensor networks," Lecture Notes in Computer Science, vol.4159, pp.595–604, Sept. 2006.

[9] H. Dubois-Ferrière, D. Estrin, M. Vetterli, "Packet Combining in Sensor Networks," Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, San Diego, California, USA, November 2005.

[10] S. Mukhopadhyay, D. Panigrahi, S. Dey, "Data Aware Low Cost Error Correction for Wireless Sensor Networks," Wireless Communications and Networking Conference (WCNC), Atlanta. GA. USA, March 2004.

[11] G. Khanna, S. Bagchi, Y.S. Wu, "Fault Tolerant Energy Aware Data Dissemination Protocol in Sensor Networks," Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN'04), Florence, Italy, June 2004.

[12] R. Stann, J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," The 1st IEEE International Workshop on Sensor Net Protocols and Applications (SNPA), Anchorage, Alaska, USA, May 2003.

[13] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, "PSFQ: a Reliable Transport Protocol for Wireless Sensor Networks," Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, September 2002.

[14] S. Biswas, R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Philadelphia, Pennsylvania, USA, August 2005.

[15] J-Sim: Java Simulator, Available at http://www.j-sim.org/.

[16] S. Madden, M.J. Frinklin, J.M. Hellerstein, W. Hong, "TinyDB: An Acquisitional Query Processing System for Sensor Networks," ACM Transactions on Database Systems, Volume 30, Issue 1, pp. 122-173, March 2005.

[17] J. Gehrke and S. Madden, "Query processing in sensor networks," IEEE Pervasive Computing, Volume 3, Issue 1, pp. 46-55, January 2004.

[18] L.M. Feeney, M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," Proceedings IEEE INFOCOM 2001, Conference on Computer Communications, Volume 3, pp. 1548-57, Piscataway, NJ, USA, April 2001.