Science Arena Publications
Specialty Journal of Engineering and Applied Science
ISSN: 2520-5943
Available online at www.sciarena.com
2018, Vol, 3 (3): 12-21

Reduction of Release Delay in the Emergency Data GTS Algorithm in Wireless Sensor Networks

Alireza Froozani Fard[1]*, Hossein Royat[2]

[1]MSc, Department of Computer Engineering, Meybod Azad University, Yazd Province, Iran,
[2]MSc, Department of Computer Engineering, Yazd Azad University, Yazd Province, Iran.

***Corresponding Author**

Abstract: *A set of wireless sensor network, composed of a number of sensors and cluster-head nodes and sink nodes, make decisions and implement them for the set of information collected from the environment using wireless communication. In present study, by entering the areas of wireless sensor networks and the GTS allocation, the emergency data GTS (EDG) algorithm, which programmed based on normal and emergency data, is investigated and a solution to the release delay improvement is proposed. The EDG algorithm has disadvantages in the area of emergency data transmission under the critical conditions: it causes an increase in the release delay in the sensor network. Thus, in present study, by implementing an appropriate threshold level and prioritizing the nodes of the first cluster-head, the high traffic of the network under the critical conditions can be controlled and the main release delay of the network can be reduced. Reduction in release delay results in reduction of the loss of emergency data in the network and increase of the network efficiency under the critical conditions. Finally, the comparison graph of the release delay and the main algorithm is investigated and the results are summarized.*

Keywords: *Wireless Sensor Network, Release Delay Improvement, Emergency Data GTS Algorithm, GTS Allocation*

INTRODUCTION

The wireless sensor network consists of very small sensor nodes that are used in the network to produce and redistribute the data. Each node consists of several sensors, a processor, a transmitter and a receiver.
Through distributed sensors, the data obtained from a standard physical phenomenon are sensed. Through the processor, sensor nodes are able to program better than what they see to do the complex task of the data transfer. The transmitter and receiver provide a wireless connection for communication. Sensor nodes have fixed and limited battery capacity. Thus, although the nodes are fixed, their topology dynamically changes regarding the energy consumption. To save energy, the node switches off the transmitter and the receiver. In this dynamic environment, the main challenge is to reconnect to the network using minimal energy. Therefore, the lifetime of the network depends on the sensor battery. Given the recent advances in the fields of integrated circuit technology, wireless communications and digital electronics, it has been possible to build small, multi-purpose, energy efficient, low-cost sensor nodes that can provide wireless communication at short distances. The high capabilities of these small sensor nodes, including measurement, data processing and communication, are very effective in realizing the wireless sensor networks according to the activity and interaction of sensor nodes (Akyildiz et al., 2010).

Wireless sensor networks

A wireless sensor network is a wireless network consisting of a large number of very small devices called sensor nodes. Sensor nodes are generally equipped with sensing, processing, and communication capabilities. Sensor nodes are distributed spatially and measure the conditions of their surroundings. The main task of the sensor node is to collect data points at regular time intervals and transform them to an electronic signal and release the signal to the sink nodes and then, to the base station through the reliable wireless communication medium. An example of the structure of a network is shown in Fig. 1.

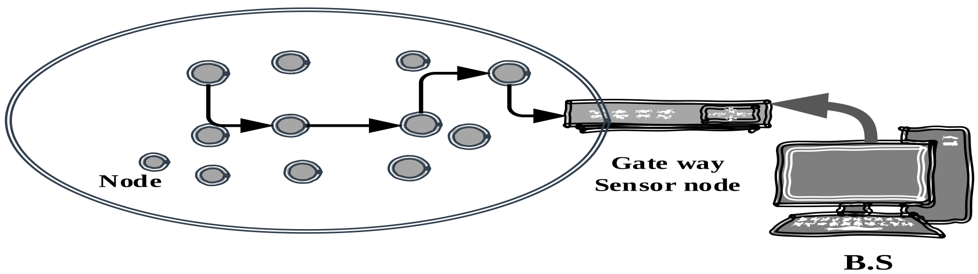

Figure 1: Wireless sensor network

Applications of Wireless Sensor Networks

The advent of wireless sensor network has led to a wide range of research in various fields. A wireless sensor network may include various types of sensors including seismic, magnetic, thermal, visual, infrared, acoustic and radar ones. They are able to monitor a wide range of environmental conditions such as temperature, humidity, pressure, speed, direction, movement, lighting, soil, noise level, existence or absence of object and the level of mechanical pressure exerted on the object. So, such a network has a wide range of applications. This range of applications includes monitoring of soil and water, collecting information for defense, environmental monitoring, urban warfare, weather analysis and forecasting, monitoring of the battlefield, exploration of solar system, monitoring of the earthquake magnitude, pressure, temperature, wind speed and geographical location information. These ever-increasing applications are categorized into five military, environmental, healthcare, home and industrial groups (Jianxin et al., 2011).

- **Healthcare and home applications**

Development of medical equipment and the advent of intelligent integrated sensors have resulted in the use of sensor networks for possible medical applications. In this regard, some applications of sensor networks are including provision of the interface for the disabled, patient monitoring, disease diagnosis, prescription of drugs in hospitals, monitoring and diagnosis of human physiological data, supervision of physicians and patients inside the hospital. The artificial retina projects, patient monitoring and emergency response can be mentioned as the projects carried out using the sensor networks.

Given the advancement of technology, intelligent sensor nodes and industrial actions can be used in home appliances such as vacuum cleaners, microwave ovens, refrigerators and freezers, DVD players, etc. Sensor nodes can interact with each other and with an external network via the Internet or satellite. They allow end users to easily control home appliances locally and remotely. A water monitoring project can be mentioned as the projects carried out in this regard (Malfa, 2010; Buratti et al., 2009; Akyildiz et al., 2010).

IEEE 802.15.4 standards

IEEE 802.15.4 is a LR-WPAN standard. It was designed for applications such as low data rates and low power consumption. This standard supports the physical layer and the MAC layer. The physical layer is responsible for the low-level operations, such as data expansion, transfer of bits, and receiving of the data regarding the interference factors, etc. On the other hand, the MAC layer creates information packets and determines the destination of data in the state of transferring. Therefore, it determines the transfer source in the receiving state and decomposes the received data. It also provides channel access control and implements multiple access protocols of the channel. Lower layers are responsible for providing data for upper layers (Aghili, 2011).

GTS allocation in IEEE 802.15.4 standard

The data transmission problem with the FCFS strategy (First Come, First Service) of the GTS allocation becomes complicated due to the lack of scheduling flexibility in meeting the applied needs and high workload of network in delivering data with low delay. For devices with low data transmission frequencies, there is a lack of data transmission because of the timer fixed and kept in 802.15.4 for capturing GTS. How to create the GTS allocation properly and adequately is a challenging issue. To solve this problem, Zhengetal have carried out the feasibility of using 802.15.4 standards on inclusive networks and Timmonsetal has analyzed the performance of the medical sensor body area networking context.

There has been a lot of research on the 802.15.4 standards performance analysis, and also, in a little number of them, GTS allocation problems have been addressed. It should be noted that the low energy consumption of wireless devices in the 802.15.4 standard and narrow bandwidth of 802.15.4 network compared to the 802.11 standard bandwidth, existing sampling algorithms (question, voting) cannot be used for GTS allocation. In Figure 2, the structure of the GTS allocation with the coordinator is presented (Demirkol et al., 2006).

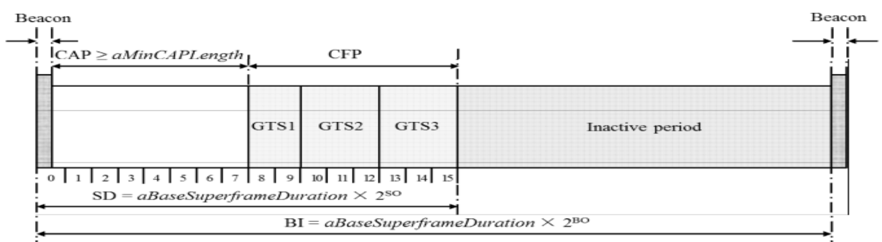

Figure 2: GTS allocation structure

- **Explicit GTS allocation algorithm**

Chen Luis et al. have proposed a design called EGSA (Explicit GTS Scheduling Algorithm), which guarantees the delay barriers in scenarios, where networked devices occupy all the seven GTS slots. EGSA also maximizes available bandwidth for applications in which time is not important). Essentially, in EGSA, CFP time slots, called small time slots, are defined according to applications. In this way, the bandwidth allocated does not losses, resulting in greater bandwidth usage in EGSA compared to the IEEE 802.15.4 standards. The above action is carried out under two conditions:

1. The number of nodes requesting GTS should be more than seven nodes.
2. The respite of each GTS message must be less than the signal time lag. The principle is that the interruption length in 802.15.4 802.15.4 during CFP must vary according to the characteristics of the message switching so that the length of each message is exactly equal to the length of the new time lag.

In the first step, the time lag is considered small, and in the next step, the signal length and CAP are calculated. Then the size of superframe order (SO) is calculated. Then, the GTS blocks are allocated to the requesting devices, and finally the length of the CAP is adjusted (Rao et al., 2013).

- **GTS allocation scheme for emergency data transmission in tree-cluster WSNs (EDG)**

Hyeopgeon Lee et al. have proposed the GTS scheduling pre-allocation mechanism for reducing the number of control packages obtained from CSMA/CA. In this algorithm, the GTS allocation program was presented for the emergency data transmission in cluster-tree topology networks. The proposed program will improve the transmission delay and output of the emergency data package. For the emergency data transmission, the EDG is allocated to the GTS slot, if the nodes send the emergency data to the coordinator, the emergency type in the GTS request frame is shown as 1, and the request of the node should be answered in the shortest time and, if the data are normal, the EDG field is zero. In Figure 3, the GTS request frame for the EDG algorithm is displayed (Der-Chen et al., 2012).

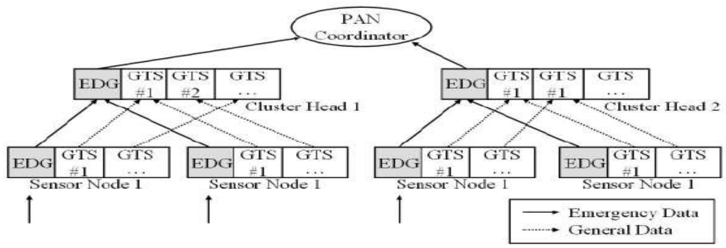

Figure 3: The GTS request frame for the EDG algorithm

- **Problems of the EDG algorithm caused by the 802.15.4 standards**

The above algorithm is developed in such a way that if the emergency data fields is 1, the data will be sent to the coordinator to be examined without stopping and with small delay, but despite some solutions suggested for reducing the delay and energy, some cases were not anticipated in the algorithm:

- If the nodes or sensors are responsible for the selection of EDG in the sensor network, then the nodes cannot wait at any time, causing the node to use the EDG field regardless of its data preference and express its data as emergency.
- If the data are incorrectly or deliberately assumed emergency by nodes, a large amount of emergency data will be sent to the cluster-head nodes of the first level and sink nodes, causing higher network traffic and delay.
- In the event of high traffic in the range of sink nodes, it will be necessary to lose the emergency data due to the termination of the package lifetime. The queuing and increasing the steps in the above cases will result in the end of the package lifetime. Thus, it creates a challenge for the network (Rao et al., 203).
- **Proposed algorithm**

If it is assumed an implemented sensor network has a cluster-shape structure as shown in Fig. (4):

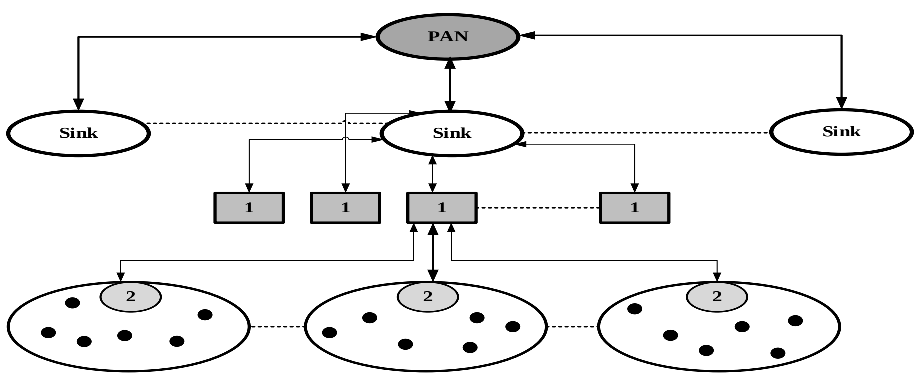

Figure 4: An example of a wireless sensor network with a cluster structure

As shown in the above figure, the network is controlled by a main coordinator named PAN. The coordinator is connected to the network environment consisting of a number of sensors or nodes through the following path: first, it is connected to a number of sink nodes, and then, to the first level cluster-head nodes and then, to the second level cluster-head nodes.

In the EDG algorithm, if the node has emergency data and requests the coordinator, the coordinator gives the GTS to it so that it can send the data and receive the response, if the entire network follows the normal mode and the nodes correctly send their emergency and normal requests, the network will not be challenged. Lets consider the conditions under which the nodes do not do their task correctly or have errors, and introduce normal data as the emergency data to the network. The more sensitive conditions are those under which the network is actually in an emergency mode and the number of actual emergency data is also increasing, and at this time, there are nodes requesting their normal data as the emergency data. Certainly, in this case, there is a high traffic in the first level cluster-head nodes and sink nodes, this increase the release delay and, also removal of the actual emergency data may occur, and some other things not mentioned by the designers. Because of all of these, the network will be challenged. Now, to resolve this issue, prioritization and/or queuing and threshold setting at the cluster-head nodes level can be proposed.

The above-mentioned problems can be addressed by managing the coordinator in the first level cluster-head nodes. If it is tried to control a high traffic in the sink nodes by prioritizing the first level cluster-head nodes, the problem of increasing the release delay will be somewhat resolved.

Prioritization of first level cluster-head nodes should be done according to the previously-described background by the sink node so that the sink node initially considers a time for the prioritization of the cluster-head nodes, i.e. it collects all the requests and the number of requests of first level cluster-head nodes in a pre-designed algorithm to separate those first-level nodes having the highest frequency of request. These requests can have normal or emergency information. When the top requesters are determined, the first level cluster-head node with the highest frequency of requests is placed in the first priority and other cluster-head nodes are placed in the next priorities according to their frequencies of requests. At this time, in the critical conditions, those cluster-head nodes having higher priorities, receive greater bandwidth to do more requests because if a set of nodes already has the highest request, it is surely in a more sensitive place when a network is in critical conditions and have more important emergency data that are more important to the network.

The pseudo-code presented in the proposed method for the release delay is shown in Fig.5:

```
%Delay to first level clusterheads
DTFLC = 0;
num = 0;
for i=1:1: n
   if(S(i). E >0 && S(i). type == 2)
      DTFLC = DTFLC + S(i). DDis;
      DTFLC = DTFLC + S(S(i). DId). DDis;
      DTFLC = DTFLC + S(S(S(i). DId). DId). DDis;
      num = num + 1;
   end
end
if (num > 0)
  DTFLC = DTFLC / num;
else
   DTFLC = 0;
end
```

Figure 5: The pseudocode of the proposed algorithm

- **Weight calculation criteria**

In simulation, it was assumed that each node has a transfer rate (Rc). Those nodes at the distance of Rc from the base station are selected as sinks, and the nodes at the distance of Rc from the sink nodes are selected as the first level cluster-head nodes. However, a weight-based distributed clustering algorithm is used to select the second level head-cluster nodes and its members.

For clustering, a weight is calculated for each node, that node with the highest weight among its neighbors is selected as the cluster header. There are four criteria for calculating the weight of each node:

1) Battery remaining: The battery remaining (Bv) of the node is the amount of remaining battery after sending and receiving data packages and messages. The higher the battery power of the selected cluster, the longer its lifetime will be.
2) The degree difference: The degree difference for each node, such as v, is calculated using the following equation.

$$N_v = |d_v - \delta| \tag{1}$$

Where d_v denotes the degree of the node v, which is actually the neighbors of the node v (the nodes inside the transfer domain of the node v), and δ denotes the number of nodes that a cluster head node can ideally deal with them.

3) Sum of distance: Sum of distance (dv) is the sum of the distances between node v and its neighbors. When the distance between a node and its neighbors is smaller, the amount of battery consumed to send messages in each node, which is dependent on the distance between the member node and the cluster header, also decreases. This will result in reduced battery consumption and increased network lifetime.
4) Distance from first level cluster headers (DFFLCH): The cluster head node that its distance from the first level cluster headers is large, will use a large amount of battery to send packages. Therefore, by choosing nodes with shorter distances from the first level cluster headers, battery consumption can be reduced in clusters (Abdel Rahman et al., 2010).

- **Normalization of Criteria**

The weight calculation criteria must be normalized after calculation, the reason for normalization is that the values of these criteria are in different ranges and all the values must be in a certain range. This normalization is considered for the range [1, 0].

$$Br_v = \frac{Br_v}{max\_power} \tag{2}$$

Where max_power is the maximum battery that a node can have.

$$N_v = e^{-N_v} \tag{3}$$

$$D_v = e^{-\frac{D_v}{d_v * Rc}} \tag{4}$$

Where Rc denotes the transfer rate of each node and d_v is the degree of node v, which is actually the neighbors of the node v.

$$DFFLCH_v = e^{-\frac{DFFLCH_v}{max\_dist}} \tag{5}$$

Where max _dist is the maximum possible distance b one node to the base station, given that the base station is assumed to be located at the center of the network, so this distance is equal to:

$$max\_dist = \sqrt{(X)^2 + (Y)^2} \tag{6}$$

Where (X, Y) is the length and width of the environment.

After normalizing, the weight of each node is calculated considering these criteria and using the following formula. The coefficients C_1, C_2, C_3, C_4 are the weight factors for the corresponding parameters of the system, the sum of which is 1 ($C_1 = 0.3$, $C_2 = 0.3$, $C_3 = 0.3$ and $C_4 = 0.1$).

$$W_v = C_1 * N_v + C_2 * D_v + C_3 * B_v + C_4 * DFFLCH_v \quad (7)$$

After selecting the second level cluster-head nodes and the member nodes, the members sense the environment and send their data to their cluster headers. The second level cluster-head node aggregates their data after receiving and sends them to its first level cluster header. There are two types of messages, emergency and non-emergency. If the message of at least one of the member nodes or the cluster header is emergency, the aggregated and sent message will also be emergency (Abdel Rahman et al., 2010).

- **Different states of prioritization**

In the first level cluster-head nodes, the messages are sent to the sink nodes after being received. In the sink nodes, the messages are queued in the order of their arrivals, if the first level cluster head node i sends a message and the queue of the sink nodes k is empty, then the message is not in the queue, but if the queue is full, there are three states:

1) If the message is normal, it will be deleted.
2) If the message is emergency and the queue has a normal message, the normal message will be replaced by the emergency message.
3) If the message is emergency and the queue does not have a normal message, the number of emergency messages sent to this first level cluster head node will be compared with the other cluster head nodes of the subset of sink node k. If the number of messages is greater, the message related to the cluster head node previously sent less emergency data is deleted and the emergency data of the cluster header i is replaced. Otherwise, the message of the cluster header i will be deleted.

- **Determination of the threshold level**

In the simulation, two types of threshold levels are considered. If there is no match with these two level, prioritization or queuing is performed, these two steps are:

The program is designed in such a way that when the operator determines a range, if the number of nodes of a designed sensor network is considered less than the threshold level, the program will not perform the prioritization, because the limited number of nodes do not need to be prioritized, and package traffic is very low. If it exceeds the range, the nodes will perform prioritization. The selected threshold level is less than 50 nodes.

After selecting the second level cluster head nodes and the member nodes, the member nodes sense the environment and send their data to their cluster head node. The second level cluster head node aggregates the information received (if the message of at least one-member node or cluster header is emergency, the aggregated and sent message will also be emergency) and sends it to its first level cluster header. There are two types of messages: emergency and normal, so if at least one data is emergency, prioritization will be carried out, otherwise the program will not do any prioritization for normal data, so prioritization is carried out for only emergency data.

Review of the simulation results

The EDG algorithm and an improved algorithm are implemented in an environment of 200 × 200 m^2 using a MATLAB simulation software and the results are compared in terms of the release delay. At first, the two algorithms are compared in the amount of data sent to the sink. Then, the release delay will be shown in the algorithm.

Simulation results

Comparison of the number of emergency and normal packages sent to the sink nodes

Fig. 6 shows the comparison of the number of emergency and normal packages sent from the first level cluster-head nodes to the sink nodes:

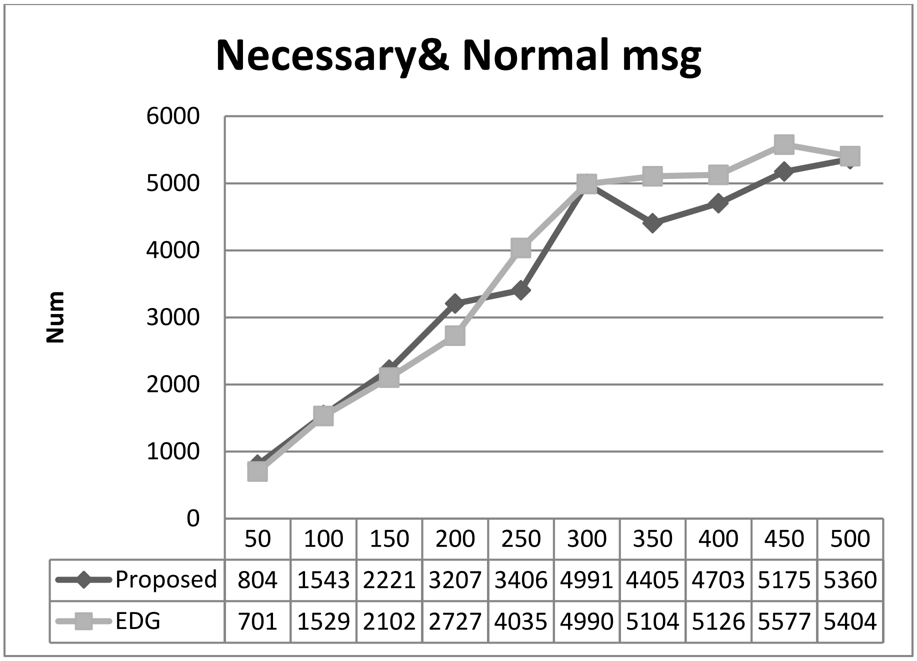

Figure 6: Comparison of the number of packages sent to sink nodes

In Fig. 6, the number of emergency and normal packages sent from the first level cluster head nodes to the sink nodes are presented. The vertical axis in the above figure represents the number of emergency and normal packages sent to the sink nodes, and the horizontal axis represents the number of rounds or harvesting of the nodes from the physical environment. The values received from both algorithms are represented below the horizontal axis. In the transmission of packages to the sink nodes, it is important that in the EDG algorithm, the values shown in Fig.6 are emergency and normal data, but in the improved algorithm, due to the prioritization, the values shown are just emergency data. So, it is seen that the amount of data in the algorithm EDG increased in the higher rounds compared to the improved algorithm. This increase in data volumes will increase the release delay, shown in Fig. (7). In the improved algorithm, due to the prioritization of the first level cluster-head nodes, greater bandwidth will be allocated to the cluster head nodes with higher priority, so the data volume will generally be reduced to avoid traffic, and only the emergency data will be transmitted to the sink nodes. Although normal data are important, but due to the high traffic, it is better to remove the normal data instead of emergency data. As in the example, in the round 350, the number of packages sent to the sink node in the EDG algorithm are 5104 and in an improved algorithm, it is 4405, indicating a high traffic in the EDG algorithm during the higher round.

Comparison of the release delay in sending the packages

In Fig. 7, the comparison of the release delay in sending the packages from the second-level cluster head node to the sink node is shown:

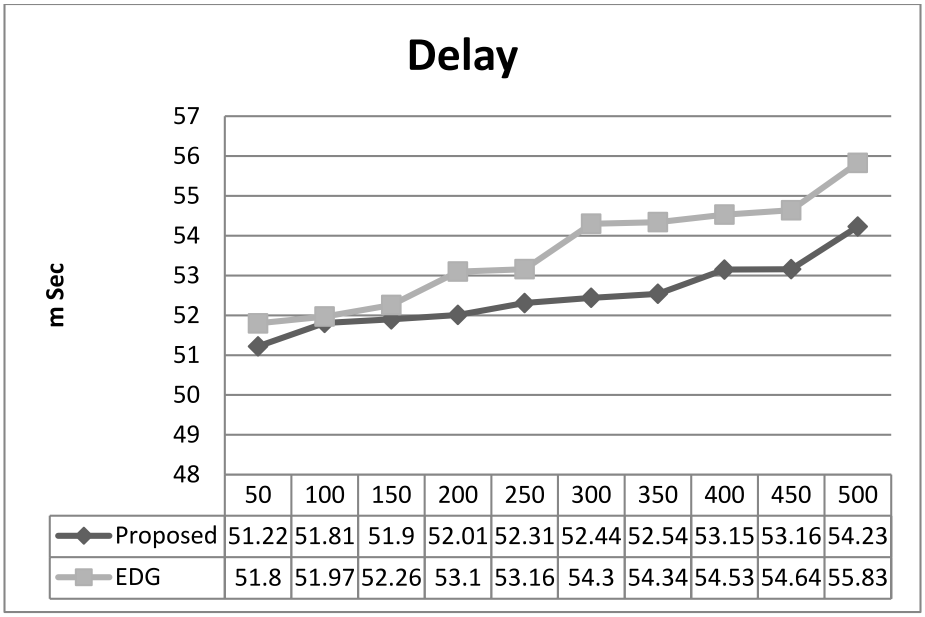

Figure 7: Comparison of the release delay

In Figure 7, the two algorithms are compared in terms of release delay. The vertical graph represents the delay in milliseconds, and the horizontal axis represents the number of rounds or harvesting from the physical environment by the nodes, the values of both algorithms are observed below the horizontal axis. Since in the second level cluster-head nodes, the operation of aggregation is performed, therefore, the delay is calculated for the first and second level cluster-head nodes and sink nodes. In fact, the delay is calculated for all nodes except for the nodes of the second level subset and then, the average is calculated. It is assumed that we want to calculate the delay of the messages sent to the second level cluster head node i. In this case, when i is sent from the second level cluster head node, the clock defined in the code starts to work and when the package is received, the clock and close time are calculated. These items are calculated for both algorithms. Calculation of the delay can be done through the calculation of the distance, but because of the same path in defining a parameter in the algorithm, this is neglected. As an example in the EDG algorithm, the delay in the harvesting of 300 from the environment is 54.3 milliseconds and in the improved algorithm, it is 52.44 milliseconds, indicating improvement due to the correct use of prioritization.

Conclusion

As noted in the present study, one of the issues that is important in the issue of release delay and its reduction in sensor networks is to reduce traffic levels and sending and receiving of the data because each node or sensor requires time to send and receive data. By reducing the release delay in the network, the response to data will increase, and this topic was discussed in present study.

In the proposed algorithm, due to the prioritization of the first level cluster head nodes based on the transmission of information from the second-level cluster head nodes, the information is queued in the sink nodes. Due to the queuing and prioritization of the first level cluster head nodes, the first level cluster head nodes, which has the highest priority, use the greatest bandwidth of sending and receiving of the data and thus the maximum GTS can be allocated to them from the coordinator. The cluster head nodes with lower priority use less bandwidth. Therefore, according to the comparative diagrams, by preventing the transmission of too much cluster head nodes with low priority, we will see the controlled traffic in the sink nodes, followed by reduced and controlled release delay, which has increased due the transmission of large amounts of information in the nodes, and decreases.

In the issue of the GTS allocation and reducing the release delay, the following example can be used as future research:

1) In the proposed algorithm, the tree topology will be used and the release delay is calculated. Other topologies can be used in this regard and the results can be compared

Reference

1. Abdel Rahman H. Hussein, Sufian Yousef, Samir Al-Khayatt, Omar S. Arabeyyat, “ A Novel Weighted Distributed Clustering Algorithm for Mobile Ad hoc Networks” , The International Journal of ACM Jordan (ISSN 2078-7952), Vol. 1, No. 3, Anglia Ruskin University, Chelmsford, U.K., 2010.
2. Aghili, Saman, Design and Simulation of a Digital Core for a Zigbee Transmitter and Receiver at 2.4 GHz for WSN Applications, Master's Thesis, Electrical and Electronic Engineering University, Tabriz, 2011.
3. Akyildiz,I,F ,& Vuran , M.C , "wireless sensor networks" , Wiley ,ISBN:978.0.470.03607.30 , 2010.
4. Buratti C, Conti A, Dardari D & Verdone R , “ An Overview Wireless Sensor Networks Technology & Evolution , Sensors” , 21 , Agust , ISSN : 1424-8220.PP.6871-6872, 2009.
5. Demirkol I, Ersoy C, Alagoz F, “MAC Protocols for Wireless Sensor Networks: a Survey”, In: IEEE Communications Magazine, 2006.
6. Der-Chen Huang, Hsiang-Wei Wu and Yi-Wei Lee, "A Cluster-Tree-Based GTS Allocation Scheme for IEEE 802.15.4 MAC Layer", Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012.
7. Jianxin Chen Luis Lino Ferreira, Eduardo Tovar, “An Explicit GTS Allocation Algorithm for IEEE 802.15.4”, IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), 2011.
8. Malfa , S,L , “Wireless Sensor Networks” , URL : WWW.doc.ic.ac.uk/project/examples 163/g1216304/references.html, 2010.
9. Rao , sharath , Keshri , Saksham , “A Survey and Comparison of GTS Allocation and Scheduling Algorithms in IEEE 802.15.4 Wireless Sensor Networks” , Students; National Institute of Technology Karnataka Surathkal, Mangalore, India, 2013.