Energy saving for wireless sensor network in manufacturing and administrative projects via Smart Nodes

A A Mahawish¹ and A S Ali²
¹Computer Engineering Department, College of Engineering, Al-Iraqia University, Baghdad, Iraq
²Network Engineering Department, College of Engineering, Al-Iraqia University, Baghdad, Iraq
E-mail: Salambash2000@yahoo.com

Abstract. The cost-effectiveness of the operation of Wireless Sensor Network (WSN) in manufacturing and administrative projects can highly boost the production of lower goods and services that in turn enhance the marketing ability. The lifetime increase of the batteries by reducing energy wastage is the main challenge and hence, in the current paper, there is an attempt to save energy via transmitting packets as quickly as can be expected while avoiding congestion and drop packets. The time synchronization allows nodes to sleep and wake at the same time to avoid consuming energy. The suggested algorithm, would, successfully deliver a packet with a fairly high rate and reduce dropping packets. Besides, it manages to avoid congestion and saves energy when reduce the retransmission the dropped packets.

1. Introduction
The Wireless Sensor Network (WSN) made from many nodes can be deployed in an environment to make sensing phenomena. This network has widely been used because of its low cost, low power, and ease of deployment. These nodes generate packets that may be continuous, based on an event or based on query then send these packets to the sink node. In WSN, the performance depends on many factors, one of them is the loss packet due to congestion. The congestion phenomena are the main reason for the loss or drop packets.

Each node has a buffer which works as store and forward transmission, temporary store packet when the output link capacity less than the number of incoming packets to avoid loss packet due to congestion. The buffer size must be managed probably because a large buffer size makes a large delay and a small buffer size will not be able to store enough packets.

Figure 1 shows the general architecture of the node in WSN. The sensing unit is small in size and can sense the phenomena and send this data to a processing unit. Then after processing, the signal needs to be transmitted or received using a communication unit to allow each node to connect with another, usually using an RF (radio frequency) channel. The node parts consume the power from the battery “limit in energy”. The sending packet needs energy as signal power to be able to send this packet [1].
There is much effort that has been done to develop the wireless communications and electronics parts over many years, which made the networks becoming cheaper, low-power consumption, and high resolution of sensors. A WSN can be constructed from many nodes that are deployed in a specific area for a specific task. The nodes work as a collaborative network to gather information from the surrounding area and then performing simple processing on the collected data. The data travel from node to node to sending it to the destination node. The main properties for a sensor network are as follows:

- Auto-organized
- Short-ranged signal communicating.
- Deployment plus cooperative effort with each other.
- Changing network topology because of node fails.
- Power provides by Battery for transmit signal, memory, sensor, and processing

Within WSN, the node broadcasts the signal to all neighbour nodes to communicate with each other. The connection topology of these nodes can be changed because of the node's fail or even shut down due to battery lack. Subsequently, the node ought to be auto organized and able to work autonomously. This type of node has some drawbacks such as energy feeding, low processing abilities, and small memory available.

A primary issue that needs to be concentrated on in WSN is their ability to be scalable [2], their association technique for connection between nodes [3], and the restricted power supply to the node. Based on the nature of WSN, applications make the battery attached with the node difficult to be replaced or recharged. Many applications domain used WSN as shown in figure 2. For example, the WSN can be used in medical fields [4] “e.g., patient observation” or even used in disaster response [5]. Also, it can be used in wildfire alarm. So, there is a requisite of managing the energy-consuming to increase the lifetime of the node. So, the dropped packet is one of the important ways to save energy for the node.

![Figure 1. Sensor Node [2].](image1)

![Figure 2. Applications of WSN.](image2)
The node buffer is a queue preceding a link that has a limited capacity. The buffer is capable of holding an infinite number of packets in a manner of the store and forward transmission. The buffer has a limited capacity, so when the sender rate is higher than the transmission rate of the outgoing link, the buffer will overflow which causes a packet drop. The queue management is used to avoid buffer overflow and to reduce packet dropping. The performance of WSN depends on several metrics [6] such as latency, energy saving, reliability, throughput, network lifetime, adaptively, and, robustness. In this study, we reduce the latency and increase the throughput to achieve energy saving as well as increase the lifetime of the network. While the issues affecting network performance are sensor node constraints, Dynamic network topology, Scalability, Heterogeneous traffic, Heterogeneous node’s processor, redundant information, Unreliable wireless channel and, congestion in the network. In this study, we focus on network congestion and the node’s processor. The Congesting occurs in nodes when the buffer overflows, which means the packet drop will happen. The Congestion is costly because the sending packet needs to be retransmitted and energy-consuming will increase. To avoid congestion, we need to implement an algorithm in WSN nodes to control the average queue size that avoids possible queue overflow.

To build and achieve our WSN network, the Zigbee [7] was used as a wireless device attached with two microcontrollers processing. The first one is PIC microcontrollers (Programmable Interface Controllers), the PIC is an electronic circuits that can be programmed to carry out a vast range of tasks [8]. The second one is ARM microcontroller, which is a series of cores optimized for power efficiency and deterministic operation. It is widely used in microcontrollers (MCUs) as a processing units and can also be found embedded into multi-core microprocessors [9]. The Zigbee uses a 2.4 GHz band that has many features such as 11 channels available, baud rate, transmission speed, and transmission power signal. Besides this section, the organization for this work can be given as follows: Section 2: presents a few previous works on congestion control and saving energy. Section 3 describes the algorithm designed for congestion control by using heterogeneous wireless nodes and how time synchronization is achieved between these nodes. Whereas, Section 4 labels the major implementing for Time synchronizing. Section 5, explains the experimentation outcomes of the Congesting Controlling Algorithm and Time synchronizing. Lastly, the findings are concluded in Section 7.

2. Relative works

Many works have been done on the main problem in WSN, which is how to improve Quality of service (QoS) and optimize power consumption. The QoS is required to meet the less packet delay, less packet drop, or increase traffic throughput, while the power consumption must be reduced to increase the lift time of the WSN node. In [10], the paper studied and reviewed the latest state-of-the-art congestion control protocols. These protocols are divided into three categories, i.e., traffic-based, resource-based, and hybrid. Traffic-based protocols are further subdivided, based on their hop-by-hop or end-to-end delivery modes. In [11], it was managed scheduling within the Medium Access Control (MAC) layer. Also, Radio Duty Cycle as an energy conservation technique was used. The design of an energy-efficient routing protocol entitled "Position Responsive Routing Protocol" (PRRP). PRRP is designed to minimize energy consumed in each node by (1) reducing the amount of time in which a sensor node is in an idle listening state and (2) reducing the average communication distance over the network [12]. The states of energy-consuming are as follows: Idle listening state: in this state the nodes use energy through many activities like sending, gathering, and waiting for a connection. They likewise consume substantially more energy through inactive and rest modes.

Collisions state: these events happen when multiple nodes send packets simultaneously. The result of the collision is dropping packets and retransmitting the required. The transmission packets will consume power, so the retransmission. Such loss in energy will occur in each node and the total energy loss will depend on the number of nodes in the path [13]. Regarding repeated transmission, strategies like arbitrary delays are important to avoid synchronize retransmit and fast return to collision event. Reducing the protocol overhead state: it indicates transmitting the packet headers plus controlling message consumes power, such information, in the end, isn't exploitable, and it ought to be
left as a minor effect. Strategies, which decrease the convention overhead by making a small transmitting interval, the prompt lesser amount of energy utilization, in turn, this also assists in subsequently sparing assets, while this increases latency [14, 15]. In outcome, a positive incentive for the transmission time frame relies upon the recurrence of progress.

3. Methodology
The suggested network of this study is created as heterogeneous networking via incorporating various Wireless types of Node such as PIC Wireless, ARM Wireless as well as PC Wireless Nodes, figure 3 below shows that.

Within heterogeneous networks there are three types of nodes, the PIC node includes a PIC microcontroller chip attached with XBee as wireless. The instruction set of PIC chip that belongs to Harvard RISC (Reduce Instructing Set Computers) and the PIC dual bus architecture give a quick and adaptable design. The PIC is attached with Wireless Device by serial cable “RS-232”. While the ARM Wireless node contains an ARM microprocessor with Wireless Device. The ARM microprocessor instruction set is also based on RISC architecture. The PC Wireless Node includes the PC connect with Wireless Device through the serial cable. These three types of nodes connect using a 2.4 GHz wireless signal. The designed network needs to apply time synchronizing between sensing nodes especially when these nodes being built up from different architectures. The time synchronization allows nodes to be active and sleep within synchronize time and this way the node be active only when its time turns on to receive the data from other nodes. The receiving node can go to sleep mode after its data sent to other nodes. So the time synchronization will participate in reducing energy-consuming in heterogeneous wireless nodes network.

There are different protocols present to apply to synchronize time, one of them was suggested by Ganeriwal et. al. which is called networking-wide time synchronizing protocols for a sensing network, Ganeriwal et. al. named that as Time Synchronizing Protocols to a Sensing Network (TPSN) [16]. This protocol has two stages of working: “level discovery stage” plus “synchronizing stage”. The former stage aims at creating hierarchic topologies within networks, as any node can be allocated a level. Level 0 is called root and only one node can be allocated, this node usually is the sink node. Regarding the later stage, the level k node synchronizes to the level k-1 node. The result at the end of the synchronizing stage, the synchronization for the whole nodes can be done to the root node, and it can achieve the networking-wide synchronizing [16].

Figure 3. Heterogeneous networking.
Implement the TPSN Algorithm:

First level discovery stage: such a stage can be operated only one time on deploying the networks. Firstly, a node ought to be resolved to be a top node “root node”. This node can represent a sink within the sensing networking. The algorithms would make synchronization for each node to an exterior period, when this sink isn’t accessible, sensing nodes could occasionally assume control over the usefulness of the root node. A current head electing algorithm may be utilized for this intermittent root node electing pace. The top-level "level 0" is allotted for the root node and then can start the level revelation stage by communicating a level disclosure packet. The node sends packets which include the identification as well as the level of this node known as the root node. When getting such packets, any receiver node assigns itself to beneath the sender node. At that point, these nodes which are beneath the sender node update the receiving packet with its identification as well as its level. When nodes are allotted levels, they dispose of additional approaching level revelation packets. Such a transmission chain continues via the networks, so the stage can be finished if the entire nodes get allocated a level.

Second level synchronizing stage: The main constructing chunk for the synchronizing stage is represented by two-way messaging exchanging amid nodes’ pairs. When considering the clock drifting amid nodes’ pairs as constant within the little timing duration through one message exchanging, the propagating delaying can be likewise presumed to be constant for the two ways.

To start to synchronize protocol, implement the two-way messaging exchanging amid node A and node B. Node A starts the synchronizing procedure via transmitting the synchronizing pulse packets within period T1 (consistent with local timing of the node). Such packet contains the level number of node A, in addition to the T1 value. Node B measures such packet (consistent with the local timing of the node) within period T2 = T1+d+Δ, as Δ represents the timing drifting among nodes, while d represents the pulse propagating delay. Node B sends an acknowledging packet within period T3, this incorporates the level of node B as well as T1, T2, along with T3 values. At that time, Node A could measure the timing drifting along with propagating delay by using the equations (1) and (2), then making the synchronization between the Nodes, from A to B [15].

\[
\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \tag{1}
\]

\[
d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \tag{2}
\]

Since the Wireless Node of this work using RS-232, so the node receives serial data only, which it waits a specific time to packetize and then to transmit the data. The packetizing timeout value is used to make the incoming packet waits for a specific time to ensure the group of packets received and then transmit again, this value is completely related to the baud rate of sending data. When packetizing value can be so small, this might lead to difficulties in which the packet needed to be transmitted together will disjoint. To test and configure the nodes, there is a need to send information as single transmitting data, yet during perform the calculating, the transmission of the initial information group is done and disjointing will occur. To solve this issue the incrementing the value of packetizing delay was used, our information could be ensured to stay jointly also to be sent within a single transmitting. The RO (Packetization Timeout) command available in XBee allows the value of packetization to change, the default value is 3, and this means that you would have to wait for 3 x (the time it takes to send all the data through serial data) before you send another packet or before you put your XBee in sleep mode. For example, I was telling you before it would be 3 x (10 ms) = 30 ms. The value of RO is set to zero which means any data "character" received by the node will send it immediately without waiting. For example, if you are transmitting a group of 10 characters and you configure the serial data to have a baud rate of 9,600 bps, that means that for one character you send through serial data would last approximate 1.04 ms per character so every data is a group of 10 bits (with start and stop bit). So
sending 10 characters in one packet needs to have 10.4 ms per packet. Now the XBee needs some time to packetizing this data and sends it all as a packet instead of sending each character every time you send it. XBee can increase such value to FF (255 in decimal).

**Congestion control algorithm:**
To set a value to the parameter of each node in WSN, these parameters are (Buffer size, Packetizing delay, threshold, Baud rate, selecting pin, register selecting, port initializing, I/O direction). The sending data starts in Wireless Node 1 to the Wireless Node 2

Let

\[ N: \text{Number of packets.} \]
\[ \text{th}1, \text{th}2: \text{the threshold to avoid packet drop.} \]

Initialize variables

At beginning the sending data without packetizing delay, so the Packetizing delay value gets set to 0. Which means that the data sending speed is high.

Loop 1: Set Packetization delay to zero

\[
\text{If (} N \geq \text{ th}1 \text{)}
\]
\[
\text{Set Packetization delay by 3}
\]
\[
\text{If (} N \leq \text{ th}2 \text{)}
\]
\[
\text{Go to Loop 1}
\]

The value of RO will affect power-consumption because the data will wait for a period of time before send it again and go to sleep mode. So, any increase in RO means increasing in power-consuming. While a decrease within the values for such parameter causes extra packet losing. The threshold \((th)\) value is used to count the average number of packets the group sends at the same time without the drop, its works like congestion controlling. The \(th\) value is determined based on link capacity and buffer size. There are two values of \(th\) \((\text{th}1 & \text{th}2)\), the \text{th}1 is used to specify the minimum number of the packet to send in one time (as window size in transport protocol), to best utilize the available resources. While the \text{th}2 that specifies the maximum number of the packet that causes sending in same time beyond that value, the drop packets will occur. When performing practice experimentations on Wireless Networking the value of the threshold can be assigned.

In experimentation practice, suppose that we have 50 packets that need to be sent. If these packets are sent as one group “by increasing RO value” within the networking, the drop packets will occur, it was obtained merely 13 packets received when sending such packets group from Node 1 to Node 2. Thus it was set threshold \(\text{th}1\) as 10 packets. After successful receiving 10 packets at Wireless Node 2, the value of RO gets incremented to 3ms and carry on sending the next group of packets, yet there is a want for saving energy by avoiding drop packets, the RO value decreasing to initial value 0 after transmitting packet group reach to 15 packets, so the value of threshold \(\text{th}2\) becomes 15. If the packets within-group are less than 15 they will be sent at high speed at one time, while if they are greater than 10 the packets will be divided from the large group into a smaller group “15 packets” in each group and reduce the sending speed to avoid the congestion. In this mechanism, we can use heterogeneous nodes and allow them to send the packet at a fair speed with avoiding drop packets.
4. Implementing
This part presents the outcomes of experimental work using the designed network as illustrated in figure 2. Such outcomes present the Time Synchronizing reading as well as the influence of the Congestion Controlling Algorithm on a network along with the way of avoiding congestion throughout communicating. The Time Synchronization algorithm was applied on the Nodes of PC, PIC, and ARM for synchronizing the time among these nodes. The outcomes can be given as follows:

Table 1. Sending and receiving time effect based on different no. of characters between two nodes.

| No. of send character | Node 1 send | Node 1 receive | Node 2 send | Node 2 receive |
|-----------------------|-------------|---------------|-------------|---------------|
| 1                     | 1           | 551           | 2           | 2             |
| 10                    | 1           | 601           | 2           | 2             |
| 50                    | 1           | 721           | 2           | 50            |
| 200                   | 1           | 931           | 2           | 330           |
| 300                   | 1           | 1101          | 2           | 490           |
| 400                   | 1           | 1261          | 2           | 720           |
| 500                   | 1           | 1421          | 2           | 930           |
| 1000                  | 1           | 2911          | 2           | 1920          |

Regarding table 1, the time in millisecond (ms) is required to send a different number of packet groups between two nodes. The $T_1$ is the time of sending a packet from node 1, while $T_4$ the receiving back the same packets to Node 1. Same for $T_2$ and $T_3$ at Node 2. The table 2 shows the time in ms for drifting and propagation delay group of character between two nodes.

Table 2. Relative Drift and Propagation delay effect based on different no. of characters between two nodes.

| No. of send character | Relative Drift | Propagation delay |
|-----------------------|----------------|-------------------|
| 1                     | 274            | 275               |
| 10                    | 299            | 300               |
| 50                    | 335            | 336               |
| 200                   | 300            | 301               |
| 300                   | 305            | 306               |
| 400                   | 270            | 271               |
| 500                   | 245            | 246               |
| 1000                  | 495            | 496               |

5. Result analysis
Now after performing the synchronization protocol and calculating the drifting between nodes, the proposed algorithm can be applied to the network. Three Wireless Nodes exist through which the data get sent (PC, ARM, and PIC Nodes). It was received 100% of packets as before the applied algorithm and there is no need to apply the Congestion Controlling Algorithm at Node 1. By using Node 2 and before using the algorithm there are only 49% packets received while after using the algorithm there
are 98% received packets as illustrated within figure 4, out of 50 packets only packet number 32 is dropped after applying the suggested algorithm, which means improving the throughput when using the algorithm on Node 2.

![Figure 4. Results for Node number 2.](image1)

Regarding Node 3 there are only 25% of packets were received before the algorithm's application, while there are only two packets were dropped which means 96% of packets got received after the algorithm submit, see figure 5.

![Figure 5. Result for Node 3.](image2)

Regarding figure 6, the chart shows the packet throughput for different types of WN “Wireless Node”. It was sent 50 packets. In Node 1 the all packets are received, while the Node 2 there are 49% dropped and the Node 3 there are 75% dropped due to congestion, which means the resend packet required and loss of energy will occur.
Figure 6. Result before applying the algorithm.

To save energy needs to reduce the dropped packet, so the suggested algorithm after applying the throughput of packets is improved and thus reduces the wastage of energy for retransmitted.

Figure 7. Result after applying the algorithm.

In figure 7, the outcome after using the congestion controlling algorithm as shown only node 2 and node 3 drops two packets out of 50 packets sent. That means the algorithm reduces the number of dropping packets and improves the performance of the network. For Node 1 it was obtained all 50 packets whereas for Node 2 it was obtained 98% and Node 3 was obtained 96%.
Figure 8. Result in percentage.

Figure 8 shows the percentage without using and using the suggested algorithm. Node 1 achieved 100% of the packets that were received in both cases. However, for Wireless Node 2 it was received just 51% of packets in the case without using the algorithm, while it was received 98% of packets in the case of using the algorithm. Similarly, it was received 25% of packets for Wireless Node 3 in the case without using the algorithm, yet it was obtained 96% of packets in the case of using the algorithm.

In this section the effect of different product are not mentioned specifically, the reason is to create network with heterogeneous node to simulate the real networks operations in the internet. The XBee as end device in each node will deal as overall with different packets processing of these products. As observation from result the XBee can control on the data rate as well as the delay to avoid congestion in the network and eliminate the drop packets.

6. Conclusions
In the current investigation, there is an attempt to save energy via transmitting packets as quickly as can be expected while avoiding congestion and drop packets. By using different type of microcontroller the time synchronization is required for allowing nodes to sleep and wake at in same time to avoid consuming energy. The suggested algorithm, as illustrated previously, delivers successfully packet with a fairly high rate and reduces dropping packets. This study show the avoid congestion and saves energy by using packetization to control on number of sending packets and the delay in queue which led reduce the retransmission packets as a result of reduce drop its.

7. References
[1] G. Anastasi, M. Conti, M. D. Francesco and A. Passarella 2009 Energy conservation in wireless sensor networks: A survey (ELSEVIER) 7(3), 537–568.
[2] B. Omodunbi, O. Arulogun and J. Emuoyibofarhe 2013 A Review of Energy Conservation in Wireless Sensor Networks (IISTE) 3(5), p. 8.
[3] M. Khedher, J. Lloret and A. Douik 2016 Enhanced group-based wireless ad-hoc sensor network protocol (UDSN) 12(7), 1-18.
[4] U. G. a. J. Bakal 2018 Healthcare Monitoring System Based on Wireless Sensor Network for Cardiac Patients Biomed Pharmacol J, 11(3).
[5] L. Parra, L. Garcia, S. Sendra and J. Lloret 2018 The Use of Sensors for Monitoring the Feeding Process and Adjusting the Feed Supply Velocity in Fish Farms (Hindawi) 2018(1), p. 14.
[6] T. K. a. D. Kumar 2020 A survey on QoS mechanisms in WSN for computational intelligence based routing protocols, Wireless Networks, 26, 2465–2486.
[7] XBee S1 802.15.4, Online Available: https://www.digi.com/resources/library/data-sheets/ds_xbeemultipointmodules, accessed on 18 February 2021.
[8] PIC18F46K22 datasheet Microchip Online Available: http://ww1.microchip.com/downloads/en/DeviceDoc/41412F.pdf, accessed on 18 February 2021.
[9] "nxp.com," LPC2131/2/4/6 2012. Online Available: https://www.nxp.com/docs/en/user-guide/UM10120.pdf, accessed on 18 February 2021.
[10] M. A. Jan, S. R. U. Jan, M. Alam, A. Akhunzada and I. U. Rahman 2018 A Comprehensive Analysis of Congestion Control Protocols in Wireless Sensor Networks (Springer) 2018(1), 456–468.
[11] M. Ojo, D. Adami and S. Giordano 2016 Performance evaluation of energy saving MAC protocols in WSN operating systems (IEEE) p. 11.
[12] N. Zaman, L. T. Jung and M. Yasin 2016 Enhancing Energy Efficiency of Wireless Sensor Network through the Design of Energy Efficient Routing Protocol (Hindawi) p. 16.
[13] B.V.Shruti, M.N.Thippeswamy and V. K 2019 Energy Efficient Medium Access Control Protocols for Wireless Sensor Networks (IJATCSE) 8(1), p. 13.
[14] A. Chugh and S. Panda 2019 Energy Efficient Techniques in Wireless Sensor Networks (Recent Patents on Engineering) 13(1), p. 9.
[15] Z. Q. Al-Khayat, S. B. Al-Maliki and A. Askar 2019 Study and mathematical modelling of the Performance indicators of Solar Tower Power Plant IJMMT, Special Issue 11(3) 29-35.
[16] S. M. Lasassmeh and J. M. Conrad 2010 Time synchronization in wireless sensor networks: A survey (IEEE) pp. 11.