An Adaptive Cross-Layer Design for Energy-Efficient Wireless Communications

Sri Sai Krishna Pradeep, Guduru
Abstract

An Adaptive Cross-Layer Design for Energy-Efficient Wireless Communications

Sri Sai Krishna Pradeep, Guduru

In this thesis we considered minimizing energy consumption with and providing QoS in WSN which is very important nowadays. In this work we developed a cross-layer protocol that operated with MAC and network layer, it introduces QoS and wake and sleep states. We evaluate to see if this new design has an effect on the energy consumption of the wireless sensor network deployment in the simulated environment. In recent years many researches have been done in WSN considering energy consumption which will increase the network lifetime. QoS, which enables reliable data transmissions by providing some guarantees, is also considered. QoS refers to the traffic controlling or traffic shaping by delivering the data in a best effort fashion, such as providing guaranteed packet delay or network throughput.

There are several energy consumption protocols being investigated like S-MAC, T-MAC, B-MAC, WISEMAC and X-MAC. S-MAC and T-MAC exchange the time schedules of data between nodes. This exchanging of time schedules is a heavy burden on the network and they reduce the life time of the network. B-MAC and WISEMAC depend on low power listening by sending long preamble that is long enough as sleep period of the receiver. This long preamble will create overhearing problem by making the non receivers to stay awake until preamble is fully received. Long preamble is a problem in target receivers because the target receiver must stay idle until the receiving of the preamble is finished, which will waste energy. In X-MAC short preamble is sent instead of long preamble and the time schedules are not exchanged between nodes which will save time and energy in the network. Because of this short preamble the source can get early acknowledgment so the data can be sent immediately and the node can go to sleep early, which is a big advantage compared to other protocols. In this work, we provide an adaptive cross-layer design for energy-efficient wireless communication in sensor networks. We modify and integrate the X-MAC protocol with random rerouting in the network layer to provide QoS in data delivery, while minimizing the energy consumption. We evaluate the performance of our design by extensive simulations in OMNET++ in terms of packet delay and energy consumption.
Acknowledgments:

My thanks first goes to my supervisor Edith Ngai for suggesting valuable comments, helping to structure the work and the presentation and allowing me to take decisions concerning the project direction. My colleagues Marco Zimmerling and Gregory Humber also helped me a lot during the project answering many questions and suggesting best ways to get good results.

I would also like to thank my reviewer Christian Rohner for valuable comments and my previous reviewer Björn Victor for his support during the project.

Finally, I would also like to thank everyone else that helped me on the thesis, particularly my friend Alexandros Tsourtis that provided valuable comments to my early version of the thesis report and also by giving some good ideas in the project. I would like thank Vanitha for the help she has done for me in bringing my papers. I would like to thank my parents that supported me during the project.

To Vanitha.
# Table of contents

1. Introduction........................................................................................................7

2. Related Work.....................................................................................................10

3. Cross-layer Design..........................................................................................13

   3.1. Neighbor Nodes..........................................................................................14
   3.2. Routing Table.............................................................................................15
   3.3. QoS.............................................................................................................18
   3.4. Energy Consumption..................................................................................22
   3.5. Collision.....................................................................................................25
   3.6. Algorithm....................................................................................................26

4. Evaluation.........................................................................................................28

   4.1. Simulation Settings.....................................................................................28
       4.1.1. Parameters...........................................................................................29
       4.1.2. Values..................................................................................................29
   4.2. Results.........................................................................................................30
       4.2.1. Energy Consumption..........................................................................31
       4.2.2. With and without Randomized Re-Routing......................................36
       4.2.3. Packet Delay.......................................................................................37
       4.2.4. Transmission Power and Collision.....................................................41

5. Future Work.....................................................................................................45

6. Conclusions.....................................................................................................46
List of Figures

Figure 1: Showing how X-MAC's short preamble is better than low power listening (LPL). Figure from [7].
Figure 2: Cross-Layer Design Framework.
Figure 3: Showing the neighbor nodes, node 4 contains no node in its range.
Figure 4: Routing of all nodes based on location. Node[4] increased its transmission range and found the sink in its range.
Figure 5: showing packet with priority 1 going from node 1 to node 3 (closest node to sink when compared to node 2).
Figure 6: showing packet with priority 2 going from node 1 to node 2 (farthest node to sink when compared to node 3).
Figure 7: Handshaking between two nodes.
Figure 8: Flow chart of the Algorithm.
Figure 9: Network topology.
Figure 10: Network with 90 nodes and sink node.
Figure 11: Difference in Energy consumption for network with sleep state of 10ms and without sleep state based on distance from sink.
Figure 12: Difference in Energy consumption when there is no sleep state and when there is sleep state of 200ms based on distance from sink for the same network.
Figure 13: Energy consumption when there is sleep state of 20ms and not having sleep state based on distance from sink.
Figure 14: Energy consumption when there is sleep state of 5ms and not having sleep state based on distance from sink.
Figure 15: Energy consumed using Randomized Re-Routing and without Randomized Re-Routing.
Figure 16: Energy consumption with different data rates
Figure 17: Delay based on different data rates
Figure 18: Delay for all nodes based on priorities
Figure 19: Transmission Power required for each node when the priority is one
Figure 20: Transmission Power required for all distances.
Figure 21: Number of collisions based on time using Randomized Re-Routing and without Randomized Re-Routing
1. Introduction:

Due to usage of Wireless Sensor Networks (WSNs) in many applications like battlefields etc, it became huge research topic [1, 2, 3]. Due to its usage there is a very huge need for reliable and energy-efficient data communications. Wireless Sensor Node is small device usually equipped with a wireless communication antenna, a microprocessor and sensors all getting their energy from a battery source. It has limited resources like energy, memory, bandwidth and computational speed. At the beginning WSNs were used in military applications and are recently used in traffic controlling, medical monitoring, security and surveillance applications etc [1,2]. Energy consumption and QoS in WSN are one of the important issues in this research. Reliable data delivery in WSN consumes lot of energy [5]. There will be many retransmissions and the data need to be propagated to the sink node. The nodes will have limited energy resources and this retransmissions etc., will consume lot of energy compared to energy available to the sensor node. We must use the resources in an efficient way, so that not only making use of the bandwidth efficiently but also minimizing the consumption of energy. It is also an interesting topic because sensor nodes are small devices and will run with batteries which will not last for long time if it is running continuously without stopping, so it is better to make the nodes to sleep when they are not in use. The network layer is the layer that chooses the network path which minimizes the error probability and energy consumption.

QoS apply to send data by giving priority to the data and users [6].QoS objective is to maximize the resource utilization by sharing the energy consumption. QoS in our implementation is performed by assigning priority to the packet so that the packets can reach the destination in different paths which will reduce bottlenecks and congestion. If the packet is having low priority then the node will route the packet in different direction than the shortest path. The shortest path is the path with less
nodes through which the packets travels. It can eliminate the unnecessary consumption of energy, while providing better services, like shorter packet delay for more important data [39]. The longer path in general consumes more energy than the shortest path but in situations when the nodes are close to each other then less transmission power is used. So the energy consumption will be less when the longer path is utilized. If there is no QoS it is more likely to get bottlenecks and congestion, and traffic will increase because without priority all packets travel through same node. This problem will not be solved by just increasing the bandwidth. If we have low traffic rate then by increasing the bandwidth will solve the problem. But in the situations where network is moderately busy we should apply QoS. Well known QoS protocols are the following: Reservation Protocol, Differentiated Services, Multi-Protocol Label Switching and Subnet Bandwidth Management. Reservation Protocol provide resources along the path to destination based on the destination address, protocol identifier and destination port. It is mostly used in telecommunications. Differentiated Services will provide QoS using priority queuing. Its main aim is to keep the core of the network as simple as possible just by moving the complexity from the network core and make the forwarding path simple. This will not only avoid the assumption of traffic type but also useful for both long-term and short-term provisioning from allocation policies and make the traffic in the form of best effort. Multi-Protocol Label Switching (MPLS) will make each packet to be forwarded based on the labels in the MPLS header field, so there is no need to read IP packet header. It is mostly used for ATM transmissions. With Multi-Protocol Label Switching there is no need to calculate routing process and also looking after address so the scalability of the network will be increased. It will also specify through which hops a packet will travel to destination. This shows how Multi-Protocol Label Switching is used to generate QoS in a deterministic manner. Subnet Bandwidth Management provides priority for the user and makes Reservation Protocol to be used by the high priority user than the low priority user. The legal user will get more priority than the free user.
Multi-hop routing will consume less energy in the source node than the direct communication in the case where the destination node is located in a long distance as communication to the node would require high transmission power, and will also decrease the idle listening time in the node by sending the nodes to sleep state[25,28,35,41]. Multi-hop routing can reduce total energy consumption because although the number hops increases the distance between the nodes decreases. X-MAC is a MAC protocol which is developed for duty-cycled WSN by sending sensor to sleep in regular interval of time [7]. By using X-MAC, we can save the energy in non-receiving nodes and avoid overhearing problems by sending short preambles with the target node address, and also the target node can send ACK immediately which will not only decrease energy but also per-hop latency. In our work we know locations of all sensor nodes so signal strength is an important factor that consumes more energy so the energy required to transmit a packet should be adjusted based on the distance to the destination node. For example, if the destination node is close, use less transmission power.
2. Related Work:

QoS is used in WSN in the following ways. CEDAR is the type of QoS aware protocol which uses the idea of core nodes of the network while determining the paths [3]. SAR is another type of protocol which is performing routing with QoS in the form of trees, by taking QoS metrics, energy resource on each path and priority level of each packet into consideration [6]. There are two types of approaches used in MAC layer to save energy exchanging the sleep and wake period time, and extended preamble and low power listening. Sensor-MAC(S-MAC) negotiates the time schedules by using a SYNC message [7,10,24,31,36,38]. When the receiver wants to send data it will wait until the receiver wakes up. When sending long messages S-MAC will divide the message into number of frames and send it, which will decrease the communication overhead by making the node to access the medium for very long time. This is also a disadvantage for S-MAC because the nodes with short packets must wait for very long time until the medium is free. When a node want to talk to the other node it will wait until the receiver start listening and send a RTS packet and the receiver will be allowed to use the medium since the RTS-CTS handshake is completed successfully. After the transmission is completed the sender will go to sleep to save its energy. Due to the sleep periods there is high latency. Timeout-MAC (T-MAC) was the improved version of S-MAC just by shortening the awake period when channel is idle [10,11,24,40]. The node will wait for a specific amount of time and goes to sleep if there is no event happened in that time. Because of going to early sleep the node may lose the synchronization of the listen period which is the problem of T-MAC. Dynamic Sensor MAC (DSMAC) is another improved version of the S-MAC [10,33]. It main aim is to decrease latency by adjusting the duty cycle based receivers sleep period. When the node receives the receivers SYNC period after the sleep period is shortened, the node will double its duty cycle based on the battery threshold. The main advantage of Data gathering MAC (D-MAC) is to decrease the latency by increasing the active period in the whole path in which data packet will travel which makes the sleep time dependent on the depth of the tree [10,12].
Scheduled channel Polling MAC (SCP-MAC) will schedule the listing of all nodes by synchronizing the channel polling, so that there is no need to use long preamble. Only a short wake-up tone is required to wake-up receiver [10,12]. SEA-MAC's main aim is to make all nodes follow only one time schedule so that the synchronization will be easy and same for all nodes [14]. The synchronization of the whole network will be maintained by base station. Eyes (E-MAC) is a TDMA based mechanism [15,30]. The node is achieving the time slot will get the RTS(Request to send) from other nodes to send data. If there is no RTS in the specified time slot then the nodes will go to standby mode. Lightweight MAC is based on the idea of EMAC [16]. Its main idea is to make the sleep adjustable based on the amount of traffic for making the implementation simple. TRaffic-Adaptive Medium Access (TRAMA) will make each node to exchange information with their 2 hop neighbor and data will be sent in the form of chronological order based on the schedules that are exchanged to the specific receiver [10,17]. TRAMA uses TDMA scheme for sending data. WISEMAC will make the length of the preamble to be fixed based on the sleeping schedules that are exchanged between neighbor nodes and the fixed preamble will be sent to receiver [10,18,23,32]. Zebra MAC (Z-MAC) is the combination of both CSMA and TDMA [20,29,30,42]. It uses CSMA when the contention is low and TDMA when contention is high. TDMA will be used when load increases and CSMA when the owners of the specific slots are not transmitting. Berkeley MAC (B-MAC) is a protocol that uses low power listening and long preamble sampling [19,27,37]. When the receiver started receiving the preamble immediately after it becomes awake then the receiver must stay awake until the preamble is finished in order to send ACK back. It consumes not only energy in receiver but also in sender and also suffers overhearing problem in which the non receiving nodes will also read the preamble.
Comparing low power listening with short preamble of X-MAC.

Figure 1: Showing how X-MAC's short preamble is better than low power listening (LPL). Figure from [7].

We can see in Figure 1 [7] that there is a lower amount of energy consumption by using short preamble where nodes go to sleep very soon than extended preamble. By the time that LPL sender was able to finish the sending of long preamble, X-MAC was able to send data also, so the rest of the time was saved and the node can go to sleep early. On the receiver side the node was able to read the preamble very fast and send ACK immediately so the sender will receive the early ACK. We can see in the time of extended wait time the node was able to receive the data as well.
3. Cross-Layer Design:

![Cross-Layer Design Framework](image)

**Figure 2: Cross-Layer Design Framework**

Figure 2 shows the protocol stack and illustrates the cross-layer design between network and Mac layer, where network layer uses the adaptive technique called Randomized Re-Routing which provides QoS by providing the best path for the destination and the Mac layer uses X-MAC which saves energy by making the nodes to sleep in regular intervals of time.
3.1 Neighbor nodes:

Figure 3: Showing the neighbor nodes, node 4 contains no node in its range

Nodes are static and each node will have a global view of the network. Each node will be given a normal transmission range of 60 meters. But isolated nodes can increase the transmission range in order to able to communicate. The positions are globally given in the form of (x,y) coordinates by using a small GPS receiver. Every node also contains a table with distances to other nodes in the network, so using that we can find the nodes in the range.
In figure 3, we can see that node[1] is having node[2] and node[3] as neighbors because the distance between node[2] and node[1] is 54m and the distance between node[3] and node[1] is 48.8m which are in the range of node[1].

In figure 3, we can also see that each node has a table containing its neighbors and also its positions. But for node[4] and sink node there is no table because sink node will not talk with other nodes, it will just listen, so there is no need to find neighbor nodes and for node[4] the transmission range is not enough so it cannot communicate with the other nodes.

To find the distance between two nodes, node a and node b. Let us consider the coordinates for node a and node b as \((x_a, y_a)\) and \((x_b, y_b)\) and by using Pythagorean theorem[10]:

$$\text{distance} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

### 3.2. Routing table:

After selecting the nodes in the transmission range the node will select the nodes from the table which are closer to sink than the original node and ignore the rest of the nodes as the nodes are sorted based on the distance from sink. If a node is an isolated node, that is the node which is not having neighboring nodes or has neighboring nodes but cannot reach sink by any paths, then it will increase its transmission range by 15m until it finds a neighbor node. If there is no node closer to sink node than the original node then the transmission range of the original node is increase by 15m. If there is sink node in the transmission range then all other nodes are excluded from the routing table except sink node because there is no need of handshaking with the sink node (Please refer to section 3.4 for information). Sink node does not sleep so no handshaking is required. The routing decision is taken by the node itself based on the location of neighbor and destination.
Distance table will be useful when there is an isolated node, let it be 'node a', trying to communicate with the other node, let it be 'node b', so that 'node b' can increase its signal strength to talk to 'node a' for reliable communication. The transmission ranges for nodes in figure 2 is found like node[0] is 60m. node[1] is 60m, node[2] is 60m, node[3] is 90m, node [4] is 90m, node[5] is 60m.

In Figure 4, we can see that both node[3] and node[4] increases their transmission range. Node[3] found the node[4] which is close to sink than the node[3] and node[4] also increases its range and found sink. Because of increasing transmission range or signal strength both the sender and also the receiver will increase their transmission power. We can also see that node[2] will not increase its transmission power to talk to node[3] because node[3] is in its range but node[3] will use the increased transmission power to talk to node[2] or node[4]. If node[3] is not in the range of the node[2] then node[2] also will increase its transmission power to talk to node[3].

In this thesis many nodes are placed in a small area so there are no isolated nodes and maximum transmission range for all nodes is same because no node is required to increase its transmission range to more than 60 meters.
Figure 4: Routing of all nodes based on location. Node[4] increased its transmission range and found the sink in its range.
3.3. QoS:

The goal of QoS is to provide certain guarantees to the network hosts. Using this QoS only the outgoing traffic is controlled but there will be no control on incoming traffic. In this project, we used the type of QoS called prioritization by inserting priority in the header field so based on that priority the node will select which route to take. If the priority is 1 then it will select the first node in the routing table to send packet. After the nodes are selected in the transmission range which are close to sink than the original node, the nodes are sorted based on the distance from sink in ascending order. The node that is closer to sink and in the transmission range of the original node will be the first node in the routing table. If priority is 2 then it will select the second node and so on etc. The priority levels are given from 1 to 5. If the size of the routing table is less than the given priority then it will select the last node in the routing table. The high priority packet will be traveled in the shortest path to the destination.

This increases the delay of the packets with less priority but important data will be sent reliably with less delay. In figure 4, we can see that node 1 which is having \{3,2\} in routing table, is transferring the packet with priority 1 through node 3 because node 3 is closer to sink than node 2 at the same time. We can also see in figure 5 that node 1 is transferring the packet with priority 2 through node 2. Its looks simple small network but in large networks there will be congestion if all nodes are traveling through same node and there will be loss of data.

Packet is of the form “(Packet)pk -1-to-Sink with priority 1 #29”. This tells that the packet will be traveling from node 1 to sink with priority 1 and the packet number is 29.
In Randomized Re-Routing technique if a node has more neighbors than priority level, the packets with lowest priority will be forwarded by the neighbors with lowest rankings with equal probability. For example, if a node is having seven neighbors and five priority levels then for priority level five it will randomly use neighbors five to seven in order to send the lowest priority packet.

By using the Randomized Re-Routing technique, we will select the nodes in the routing table based on the distance; if the neighboring node is closest to the sink than the original node then the node is selected. This is because if the node is far from the sink when compared to original node then it will lead to very long paths. The highest priority packets are sent through the nodes that are closest to the sink so the number of hops will be reduced and the low priority packets are sent through other nodes based on priority. This tells that the number of hops increases based on priority. This shows that Randomized Re-routing technique will make the low priority packets to travel away from the best path which will not lead to congestion by sending many packets through one path. The below figures shows how the Randomized Re-Routing technique will divert the packets based on priority.

In figure 5, we can see that a packet with priority 1 is traveling through node 3 from node 1 and from node 3 it travels to node 4 and then to sink. The dashed line shows the path from node 3 to sink.
Figure 5. showing packet with priority 1 going from node 1 to node 3 (closest node to sink when compared to node 2).

In figure 6, we can see that a packet with priority 2 is traveling through node 2 from node 1 and from node 2. Then, it travels to node 5 and then to node 6 and sink. The dashed line shows the path from node 3 to sink.
Figure 6. Showing packet with priority 2 going from node 1 to node 2 (farthest node to sink when compared to node 3).

The delay increases in the second situation compared with that in the first situation.

This is a small network for everyone to understand. In huge networks there will be many neighbor nodes for every node so there is will be many paths and the packets with different priority travels in different paths and the congestion will be reduced.
3.4. Energy consumption:

Energy consumption can be reduced by making the nodes to sleep in a regular interval of time so that there is a need of handshaking to check whether the node is in sleep state or not. There is no need to do handshaking with sink node because it will not sleep.

Note: Data will not be acknowledged in X-MAC.

Handshaking will be done in the following way:

**Figure 7: Handshaking between two nodes.**

If node[2] is in sleep state then node[1] will continuously send the preamble until it gets acknowledgment. If a node contains information then it will not go to sleep state until it sends whole information to next node even though its sleeping period occurred just by checking whether there is any data packet in the memory or not. If a node receives more than one packet then the packet is saved in the form of linked list along with the previous packet and while sending the packets, the node will send in the form of First In First Out process.
Every time when a node went to sleep or idle, time will be calculated and added to the previous sleep or idle time. For every transmission of preamble, acknowledgment and data packets, node will calculate the energy individually. The signal strength required to send to a particular node will be calculated based on the distance to the node. For example if ‘node a’ is sender and 'node b' is receiver then 'node a' will calculate the power required to transmit the packet to that particular node as follows.

Let \( P_{\text{tx}} \) be the power delivered to transmit the packet to D distance, where D is the distance between node a and node b. The power received by the destination using Frii’s transmission equation[44,45,46] is

\[
P_{\text{rx}} = P_{\text{tx}} \times \left( \frac{\lambda}{(4 \times \pi \times D)} \right)^2,
\]

where \( \lambda \) is wavelength.

The power required to transmit to destination node with distance D is

\[
P_{\text{tx}} = P_{\text{threshold}} \times ((4 \times \pi \times D)/\lambda)^2,
\]

where \( P_{\text{tx}} \) is the power required to transmit and \( P_{\text{threshold}} \) is the minimum required receiving power.

So, by reducing the signal strength the energy can be saved. The time node sleeps, idle and transmitting will be calculated in milliseconds.

The energy required to transmit preamble will be

\[
P_{\text{tx}} \times T_{\text{pt}},
\]

where \( T_{\text{pt}} \) is time taken to send preamble.

The energy required to transmit acknowledgment will be

\[
P_{\text{tx}} \times T_{\text{at}},
\]

where \( T_{\text{at}} \) is time taken to send acknowledgment.
The energy required to transmit data will be

\[ P_{tx} \times T_{dt}, \]

where \( T_{dt} \) is time taken to send data.

The energy required to send the preamble, acknowledgment and data packets for a node will not be same because node can send packets to different neighbors with different distances so the energy required to transmit packet will vary.

Let us assume that the total energy spent by a node to send Preamble, total energy spent by a node to send Acknowledgment, total energy spent by a node to send data be 'P', 'A' and 'D_a' and the total time node sleeps and total time node is Idle be 'S' and 'I'.

Then the total energy spent by a node with sleeping schedule is

\[ E_s = P + A + D_a + (S \times P_s) + (I \times P_i), \]

where \( P_s \) is the power taken to sleep and \( P_i \) is the power taken to idle or listening.

Then the total energy spent by a node without sleeping schedule is

\[ E_i = D_a + ((S + I) \times P_i), \]

where it was shown in results that \( E_i > E_s. \)

The values of the \( P_{tx}, P_s, P_i, T_{pt}, T_{at}, T_{dt} \) can be seen in Section 4.1.2.

The preamble that is being sent contains only the target ID so that when the receiver wake up and receives the preamble it will check whether the target ID is itself or not. If the receiver is the target then it will send acknowledgment and will not go to sleep until it receives the data packet and send the packet to the next node. Because of shortening this preamble the time taken to receive is also reduced. The preamble with only target ID
is the short preamble which will avoid the overhearing problem in the non target receivers. It lets the non target receivers to go back to sleep quickly so that the energy is also saved, which is a big problem in other asynchronous MAC protocols like B-MAC and WISEMAC [43]. Using this asynchronous method will reduce the burden of negotiation of sleep schedules between the nodes like in Synchronized MAC protocols S-MAC and T-MAC.

Using this handshaking is not only an advantage but also a disadvantage for this protocol because the node must wait for longer time in order to send data but this is the only way for the sender to know that the receiver is awake in asynchronous protocols. Another problem is that many transmitters are sending preamble to one receiver, so it is very important for the receiver to handle this kind of situations. After transmitting the data packet the node will remain awake for some time and then go to sleep, just in case if any other transmitter may send preamble.

3.5 Collision

When two or more packets arrives to same node at same time then collision will occur. Collision is one of the events that will consume energy of nodes because there will be retransmissions of packets and the node will stay awake more time than actually it must be[22,34]. When collision occurs all the packets that arrived to that node at that time will be discarded and the time spent in the data transmission is wasted as the channel was busy but no data were received. When the number of users increases then the collisions also increase and will result in the delay of the packet [22]. In X-MAC, when the data rate is high then collision is a problem but when data rate is low then collision is not a problem, number of collisions decreases when the data rate is low. In this thesis collision is detected but not avoided.
3.6 Algorithm:

![Algorithm flow chart]

**Figure 8: Flow chart of the algorithm**

Steps:

1. **Initialization:**

   i. Take the positions of all nodes in the network.

   ii. Find the distances with all nodes using the Pythagorean Theorem using position table.

   iii. Find the neighbor nodes using the distance table.

   iv. Using step iii find the nodes that are near to sink than the original node.

   v. Sort the nodes in step iv based on the distance to sink in ascending order.
2. Duty Cycle:

i. Make the nodes to go to sleep for 20 milliseconds if the node is idle for 20 milliseconds.

ii. When the node has a packet to send or receives preamble then do not let the node to go to sleep.

3. Communications:

i. If node has a packet to send then send a preamble to the next node based on priority.

ii. If packet has high priority then send it to the first path in routing table which has less hops else depending on the priority select the second third etc path. If priority is greater than number of nodes in the routing table then it selects the last node in routing table. If a node has many neighbor nodes then then the packets with lowest priority will forwarded to neighbors in routing table with equal probability.

iii. Go on sending preamble until it gets acknowledgment.

iv. When node gets acknowledgment then send data packet.

v. The strength of the signal to send to neighbor node depends on the distance between nodes. If longer distance then use more signal strength else less signal strength.
4. Evaluation:

4.1 Simulation Settings:

Simulation was performed using OMNET++ in the area of 600m x 400m (width x height) with 91 nodes where each node is having a transmission range of 60m and all the nodes are distributed randomly in that area. There was no isolated node in the area. The priority range is from 1 to 5 where 1 is the highest priority and 5 is the lowest priority. The device considered for experiment is Telos B mote sensor. The destination node will be the last node which is the node 90 shown as the sink node. There is only one sink node; it is located in the red box which can be seen in figure 9. In network layer we used the Randomized Re-Routing, which will provide QoS in routing and in Mac layer we used X-MAC, which will save energy and increase the network lifetime.

Figure 9: Network Topology
The parameters used and their values used are given below.

### 4.1.1 Parameters:

- \( P_{tx} \) = Power required to Transmit
- \( P_s \) = Power required to Sleep
- \( P_i \) = Power required to stay Idle or Receiving
- \( P_{pt} \) = Time taken to send Preamble
- \( P_{at} \) = Time taken to send Acknowledgment
- \( P_{dt} \) = Time taken to send Data
- \( R \) = Range of Transmission

### 4.1.2 Values:

These results are of the device Telos B mote Sensor [21].

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>60m</td>
<td></td>
</tr>
<tr>
<td>( P_{tx} )</td>
<td>172.4mW</td>
<td>Device-specific</td>
</tr>
<tr>
<td>( P_s )</td>
<td>0.0183mW</td>
<td>Device-specific</td>
</tr>
<tr>
<td>( P_i )</td>
<td>96.6mW</td>
<td>Device-specific</td>
</tr>
<tr>
<td>( T_{pt} )</td>
<td>1.98ms</td>
<td>Measured</td>
</tr>
<tr>
<td>( T_{at} )</td>
<td>1.84ms</td>
<td>Measured</td>
</tr>
<tr>
<td>( T_{dt} )</td>
<td>3.8ms</td>
<td>Application Specific</td>
</tr>
</tbody>
</table>

Results of this table are taken using Table 3. in [7]

The data rates used 1 packet/second, 2 packets/second, 3 packets/second, 4 packets/second and 5 packets/second
4.2. Results:

All results that are given below are taken by running the simulation for 1 hour simulation time with 90 nodes and 1 sink for the network shown below.

![Network with 90 nodes and sink node](image)

Figure 10: Network with 90 nodes and sink node

Note: In simulation node numbers starts from 0 to 89 and the sink node is number 90, but in below graphs node number starts from 1 to 90 and sink is node 91.
4.2.1. Energy Consumption:

![Energy Consumption Chart](image)

Figure 11: Difference in Energy consumption for network with sleep state of 10ms and without sleep state based on distance from sink.

The total energy consumed when there is no sleep state is $3.1133\times10^{10}\mu$J and the total energy consumed when there is sleep state with 10 milliseconds is $2.5377\times10^{10}\mu$J. This shows that the $\sim18.4\%$ of energy was saved because of sleep state than without sleep state. Most of the traffic that are further away from sink node passes through this nodes. It is because the nodes that are 200m distance is the gateway for the further nodes. These nodes must send their packets and also other nodes...
packets. The sink node which will not sleep and just listen or receive the packets so the energy consumed is same by the sink node when there is sleep and no sleep state. Figure 11 shows the average energy consumption of the nodes in the specified range of the distances. For example, the first set of bars is the average energy consumption of nodes in the range of 72m from sink and the second set of bars are the average energy consumption by the nodes from 72m to 144m and so on. The nodes that are near to sink consume less energy when compared to the nodes far from the sink. This is because most of the nodes in the first set of bars are having sink node in their range so there is no need of extra transmissions preambles and acknowledgments.

Note: handshaking was not considered when there is no sleep state.

Figure 12: Difference in Energy consumption when there is no sleep state and when there is sleep state of 200ms based on distance from sink for the same network.
Figure 12 shows the total energy consumed when there is no sleep state and with sleep state of 200 milliseconds. The total energy consumed when there is no sleep state is 3.1053e+010µJ and the total energy consumed with sleep state is 2.8341e+010µJ. Because of having sleep state only ~8% of the total energy was saved. The energy consumption is more when the sleep state duration increases and less when there is less sleep state. This is because the node will send preamble many times until it gets acknowledgment. We can clearly see in figure 12 the values of green bars are more in sleep state when compared to the bars in figure 11, which shows that the small sleep state is better than the large sleep state and also ~10% of extra energy was saved with less sleep state.

Figure 13: Energy consumption when there is sleep state of 20ms and not having sleep state based on distance from sink.
Figure 13 shows how the energy consumption will be when the sleep time is 20ms. The total energy consumed when there is sleep state is $2.6176 \times 10^{10} \mu J$ and the total energy consumed when there is no sleep state is $3.1104 \times 10^{10} \mu J$. Hence, the total percentage of the energy saved was $\sim 15\%$ which is less when compared to 10ms. Figure 14 shows how the energy consumption will be when the sleep state is less than 10ms.

![Figure 14: Energy consumption when there is sleep state of 5ms and not having sleep state based on distance from sink.](image-url)
Figure 14 shows the energy consumption when the node is in sleep state of 5ms and when it not in sleep state. The total energy consumed when there is sleep state is $2.7320 \times 10^{10} \mu J$ and the total energy consumed when there is no sleep state is $3.1021 \times 10^{10} \mu J$. The percentage of energy saved is only $\sim 11\%$ which is less than the percentage of energy saved with 10ms sleep state. This is because, when there is 10ms sleep time there is less probability that the node will send the preamble when the neighbor node is in sleep state. At the same time for 5ms sleep time there is high probability that the node will send preamble when the node is in sleep state because the node will go to sleep very early than it will go with 10ms sleep state. Because of this the node will send many preambles until it gets acknowledgment. We can also see in figure 11 that all nodes energy consumption was increased instead of some specific nodes like in 20ms.

When sleep state is 10milliseconds the energy consumed is $2.5377 \times 10^{10} \mu J$, when 20milliseconds the energy consumed is $2.6176 \times 10^{10} \mu J$, when 5milliseconds the energy consumed is $2.7320 \times 10^{10} \mu J$. This shows that the sleep period must not be very less and at the same must not be very high.

When the sleep time increases latency also increases.

Figure 16 shows the energy consumption with different data rates.

This figure shows that the energy consumption is very high when there is high data rate and at the same time the number of packets traveling in the will be more when there is high data rate. The green part of the figure shows the number of packets traveled. In total most of the energy is consumed by listening state. The percentage of energy consumed for each data rate in idle or listening state is shown below:

1 packets/second - 99.1%
2 packets/second - 98.5%
3 packets/second - 97.8%
4 packets/second - 97.3%
5 packets/second - 96.7%

This shows that when data rate is high, idle time will be less.
Energy consumed for sleeping is:

1 packets/second - 1.5328e+006µJ
2 packets/second - 1.1652e+006µJ
3 packets/second - 1.1584e+006µJ
4 packets/second - 1.0623e+006µJ
5 packets/second - 1.0319e+006µJ,

which is very low and can't be seen in figure 16.

4.2.2. With and Without Randomized Re-Routing:

Figure 15 shows the energy consumption when there is Randomized Re-routing (RRR) and Direct routing. The energy consumed using Randomized Re-Routing is 2.5377e+010 and the energy consumed without using Randomized Re-Routing is 2.5255e+010. We can see that the energy consumption is almost same because the lowest ranking neighbor nodes are very close to the sender and the power required to send to that neighbor nodes is almost same.

With and without Randomized Re-Routing the sender will send the packets with priority 1 to 4 to next node in the same way as mentioned in section 3.3. But when the priority level is 5, with Randomized Re-Routing and sender having more neighbor nodes than the priority level, the sender will send through different low ranking nodes instead of always sending through the same node. But without Randomized Re-Routing the packet with priority 5 will travel through same node. So when using Randomized Re-Routing, the packet with priority level 5 will travel from different nodes, there will be variation in distance between source and destination and more power will be consumed when compared with not using Randomized Re-Routing. So the total energy consumed is more when using Randomized Re-Routing than not using Randomized Re-Routing.
4.2.3. Packet Delay:

The two figures 17 and 18 are based on delay, for which there is no sleep state and handshaking. This is because if there is sleep state then there is a chance of getting low priority packet before high priority packet and that depends on the state of the neighbor node. There is no need of handshaking if there is no sleep state. Priority levels are from 1 to 5, where 1 is high priority and 5 is low priority.
Figure 16: Energy consumption with different data rates
Figure 17: Delay based on different data rates.

Figure 17 shows the delay is almost same for all data rates with difference of 0.18ms which is very small.
Figure 18: Delay for all nodes based on priorities
Figure 18 shows how the delay varies based on priorities for all nodes. We can see in some nodes that the delay taken by priority 4 is high than priority 5 (for example nodes 1,3,21,44,46,54,64,67,74,80), this is, the neighbor to which the node is sending priority 4 packets are having more neighbors and the neighbor to which priority 5 packets are sent are having less neighbors, so there is difference in delay. However, we can see that all nodes with priority 1 are experiencing less delay than all other priorities.

4.2.4. Transmission Power and Collision:

Figure 19 shows the power used by all nodes to transmit to its neighbor node when the priority is one. The green line represents the receiving or listening power so that we can say that the transmitting power is less when the neighbor nodes are very close. In this figure, we can see that almost 10 nodes are using less power than the receiving power.

For rest of the priorities the transmission power should not be more than the transmission power required when priority is one.

Please refer to section 3 for values used.
Transmission Power required for each node for priority 1

Figure 19: Transmission Power required for each node when the priority is one
Figure 20: Transmission Power required for all distances

Figure 20 shows the transmission power required for all distances. It is shown that the distances below 44.6m will use less power for transmitting data than receiving data power.

This tells that the energy can be saved based on the signal strength so the network lifetime will be increased.
Figure 21: Number of collisions based on time using Randomized Re-Routing and without Randomized Re-Routing

Figure 21 shows how the number of collisions increases when the time increases. The results are taken with high data rate almost equal to 10 packets per second. This is the reason the number of collisions are very high. It is cumulative number of collisions. The collisions occurred in 30 seconds is the collision occurred at 20 seconds plus the collision occurred at 10 seconds. That is why the graph grows exponentially.

The number of collisions without using Randomized Re-Routing is more when compared to using Randomized Re-Routing because for priority 5 packets, every node will have same path and when traffic increases the packets that will travel through that node will increase and there is chance of having congestion and collisions. When using Randomized Re-Routing the priority 5 packets will travel through different paths and there is less chance of having congestion and collisions when compared to not using Randomized Re-Routing.
5. Future Work:

Even though the X-MAC is giving better results, idle listening is consuming almost 98-99% of the total energy. It will be better to reduce it by making at least the nodes on the edges to go to sleep early and make the network core to work as it is. We can also make the nodes that are not in use for very long time to go to sleep for very long time and the nodes that are being used regularly to go to sleep for very short period of time.

When it comes to QoS, using priority is good way, it will be better if there are more neighbors. When it comes to less neighbors than the priority levels then it will be better to send packets based on the the routing table. For example, if a node has 3 neighbors and the priority level is 5, then the priority 1 packet will always be given first preference as usual. However, for rest of the priorities it will be better to send 2,3 priority packets through the 2nd neighbor node in the routing table and 4,5 priority packets through 3rd neighbor node in the routing table, so that the work will be distributed equally to all neighbors. Instead of sending all 3,4,5 priority packets through the node[3] in the routing table.

As the collision rate increases when the data rate increases, we could try to detect and avoid collisions.
6. Conclusions:

The main aim of this thesis is to reduce energy consumption in WSN by making the nodes to sleep and also introducing QoS by using priority. Reducing the transmission power based on the distance to the receiver is one of the ideas proposed in this thesis. When the neighbor node is very close to the source node then the power required to transmit will be very less to that neighbor. This is one of the ways where energy can be saved. Sometimes the transmitting power will be less than the receiving power.

The shortened preamble introduced in X-MAC with the address of the target receiver will reduce the latency and overhearing problem. It leads the node to go to sleep state early by receiving early acknowledgments, so the time the node will be in idle state will be less and as a result the energy consumption will decrease in both sender and receiver. Energy consumption in the receiver will also be reduced because the time taken to listen to short preamble will be short and the receiver can send the acknowledgment immediately to the source and wait to receive data. All this process is faster in X-MAC when compared with LPL. The nodes which are not targets also can go to sleep immediately by seeing the target address in the short preamble. X-MAC performs better than LPL (Long Preamble Listening) because of these short preambles. Prioritized QoS is simple to implement. It gives more importance to high priority packets, by making the high priority packets to travel through the network core and the low priority packets to reach destination by traveling through the edges of the network. Hence, the network core will not experience high traffic loads which will also reduce congestion in the core.

The cross-layer design in this thesis is between network and Mac layer where the network layer uses Randomized Re-Routing technique for providing QoS as mentioned above, and the Mac layer uses X-MAC for energy consumption. Consumed energy is less with the introduction of X-MAC but that causes the delay to increase more than expected. Energy consumption changes are based on the sleep time and also on the data rate of the network. This cross-layer design saves ~17.7% of energy which will increase the network lifetime.
Bibliography


