Modular Topology Control and Energy Model for Wireless Ad Hoc Sensor Networks

Niranjan Krishnamurthi, Shanchieh Jay Yang, Mark Seidman
Rochester Institute of Technology
Rochester, New York, 2004
E-mail: nxk5767@rit.edu, sjyeec@ce.rit.edu, MSeidman@mail.rit.edu

Abstract

Networked wireless sensors in a harsh terrain typically are battery operated and, therefore, require energy efficient network protocols. In order to ease the analysis of the energy usage of proposed network protocols, this paper proposes an energy module, implemented in OPNET, to determine the energy consumption over time during network simulation. Multiple energy consumption modes, such as transmit, idle, sleep modes are all accounted for. A sensor node will be automatically turned off (disabled) in the middle of simulation if it runs out of the user-specified energy. The energy module provides function calls for network protocols implemented in OPNET, and therefore allows performance evaluation in terms of the energy efficiency. Among other protocols, this paper focuses on the topology control schemes, which has individual wireless sensors determine their own transmission power levels while maintaining network connectivity. In order to allow the adjustment of transmission power level having an effect on the transmission range, this paper investigates necessary modifications on the Closure Pipeline stage in OPNET. A simple topology control scheme is implemented for validation purposes. Simulation results will be presented in this paper to exhibit the effect of the closure stage modification as well as the energy modules.

1 Introduction

Many challenges arise as the network evolution steps into the era of self-configured / adaptive wireless sensor networks. These wireless sensors, perhaps constrained by their size and cost, may form multi-hop ad hoc networks to collectively retrieve environmental information. A key factor that leads to the new challenges in performing networked sensing is the limited battery resources of the wireless sensors. Therefore, it is essential to design network protocols that achieve energy efficient retrieval of sensed data. In order to analyze the efficiency of proposed protocols, this paper proposes an energy module, implemented in OPNET, to determine the energy consumption over time during network simulation. The energy module monitors how energy is spent based on user's definition on various energy parameters, such as the power levels when operating in the sleep, idle, and transmit modes. A sensor node then will be automatically turned off (disabled) in the middle of simulation if it runs out of energy.

The energy control module provides function calls for network protocols implemented in OPNET, and therefore allows performance evaluation in terms of the energy efficiency. Among other protocols, this paper focuses on the topology control schemes [2,3,4], which is not typical for traditional network modeling. A topology control scheme has individual wireless sensors determine their transmission power to be at a level that conserve energy while maintaining network connectivity. In order to allow the adjustment of transmission power level having an effect on the transmission range, this paper investigates necessary modifications on the Closure Pipeline stage in OPNET. A simple topology control scheme is implemented for validation purposes.

The paper is organized as follows. Section 2 gives a brief overview of the node model used. Section 3 describes the energy control module in detail. Section 4 states the topology control module in detail. Section 5 discusses the simulation results whereas Section 6 presents the conclusion.

2 Overview of the Node Model

The development of the energy module and the topology control module is independent of the modeling of a wireless node. For presentation purposes, we consider a publicly available node model [1], and build our developed modules as a case study. The resulting wireless node model is shown in Figure 1.

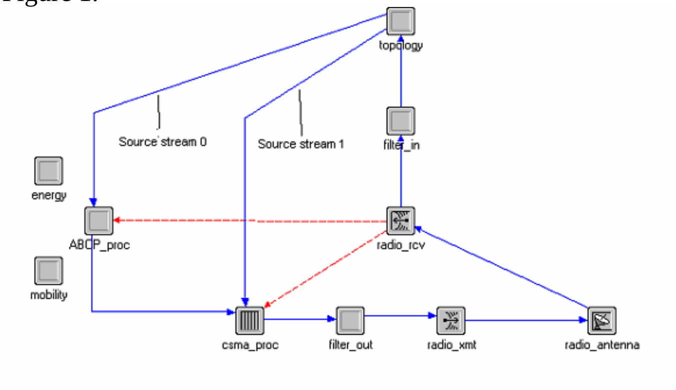

Figure 1: Node model

The model comprises a standard radio antenna, a modified radio transmitter, a modified radio receiver, a routing protocol called ABCP, a CSMA/CA MAC module, a mobility control module, two packet-filtering modules, and the energy and the topology control modules. Among the various components, the routing, the MAC, and the mobility control modules are adopted from [1]. All other modules are developed or modified and serve as contributions of this paper.

A packet received by the antenna will pass through the receiver, and "filtered" by the "filter_in" module. The filter_in module will dump the packet if the wireless sensor node is in the sleep mode or has already used up its pre-specified battery energy. If the node is alive or not in the sleep mode, the packet will be passed to the topology control module, then to the routing and MAC modules or directly to the MAC module. Whether the packet will be sent to the routing module depends on the type of packets received. If the received packet is used to determine the transmission power level, i.e., a topology control packet, then the control flow will bypass the routing protocol and directly head to the MAC protocol. Otherwise, the packet will be passed to the routing protocol to help make necessary actions. A "filter_out" module is developed before outgoing packet sent to the transmitter. Similar to the filter_in module, this filter-Out module will dump all the outgoing packets if the wireless node is in the sleep mode or has used up the energy. The existence of the filter_in and the filter_out modules ensures that the decision on the wireless sensor operational modes, which can be changed via functional calls to the energy module, overwrites other protocol's decision on sending and receiving packets.

3 The Energy Control Module

The purpose of the energy module is to monitor the energy consumption over time of each wireless sensor node. Given a user specified amount of total energy available to a node, the energy consumed depends on the operational modes the node is in. We consider three operational modes: transmit, idle, and sleep. Note that researchers have also considered a "receive" mode in addition to the three aforementioned modes. The energy consumption rate for receiving a packet is, however, reportedly not very different from simply keeping the sensor's wireless communication circuitry "on", i.e., that consumed in the idle mode. We therefore omitted the modeling of the receive mode, and it shall be straightforward to add a receive mode to our proposed model.

The energy consumption rate, i.e., the power level, for the transmit mode is calculated based on the distance of the neighbors, the transmission capacity, and the size of the message to transmit. The users can specify the power level for the idle and sleep mode. The energy module then will track the usage of the energy until the remaining energy becomes zero. Once the user-specified initial battery energy has been used up, the energy module will "fail" the transmitter and the receiver of the node. The various energy usages over time will be collected and can be selected to show as the simulation statistics.

The process model (i.e., the finite state machine) of the energy module is shown in Figure 2. The parameters that can be defined by the user at the beginning of the simulation are summarized in Table 1 along with their default values. The specified values will be initialized in the "Init" state in the process model. Once initialized, the process model will be at default remain at the "Idle" state. Periodically, the process model will issue a self-interrupt to go to the "Idle_consume" and calculate the energy consumed due to staying in the idle mode. If a external interrupt is issued by any other network protocol modules, such as the topology control or the MAC protocol, the finite state machine may transit from the Idle state to the "Consume" state or the "Sleep" state. External processes shall invoke the transition to the Consume state if and only if any message is ready to be sent. If the node has enough remaining energy to consume the required energy, then the energy to be consumed is subtracted from the available energy and a state transition is made back to the idle state. If an external interrupt is issued to make the node to "sleep", then the energy module will be instructed to dump all incoming and outgoing messages by using the filter_in and filter_out module discussed in Section 2. In the "Sleep" state, a user specified power level would be used to keep track of the remaining energy. If the node does not have enough remaining energy in any of the Idle_consume, Consume, and Sleep states to perform the current task, a state transition will be made to the "Dead" state, and the wireless sensor node will be "disabled."

Description	**Default Value**
Initial node energy	25000 Joules
Idle consumption rate	0.005 Joules/second
Sleep consumption rate	0.0005 Joules/second

Table 1: User specified energy attributes.

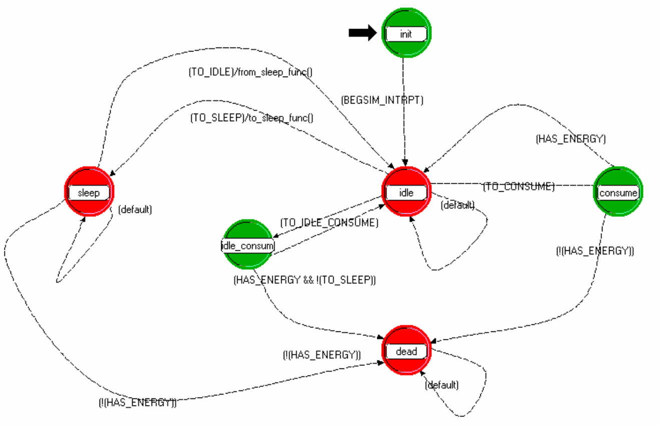

Figure 2: The process model for the energy control module.

4 The Topology Control Module

Topology control for wireless ad hoc sensor networks has drawn increasing interests in the literature [2,3,4]. One way to establish the network topology is to control the transmission power level of each node. This in turn determines whether the packet sent by a node can successfully reach the neighboring nodes, based on the distance to those nodes. OPNET, however, does not provide functional calls to change the transmission power during the simulation. Therefore, changes were made to OPNET pipeline stages to allow for dynamically adjusting the transmission power, so that topology control protocols may be implemented and tested. The following criterion is used in our model to determine whether the packet sent by node A could reach node B. Define the transmission threshold as

$$r_{ab} = \frac{d_{ab}}{p_a^{\alpha}} \tag{1}$$

where d_{ab} is the Euclidean distance between the node A and B, and p_a is the transmission power level of node A. The parameter α in (1) typically ranges between 2 to 4 to reflect the signal attenuation over distance. The default value of α used in our model is 2, but can be changed by users in the header file. The packet sent by node A can reach node B if and only if $d_{ab} \geq 0.001$. The default value of 0.001 can also be changed in the header block of the pipeline stage. Notice that this change to the pipeline stage is amended to the existing code, and will not alter the impact of the line of sight and the altitude of the nodes. For example, a transmitter with a transmission power of 1 W will have a transmission range (assuming a proper line of sight) of 31.62 meters by easy calculations.

At the beginning of the simulation, the node sends a “help” packet and waits for a reply. If no reply is obtained within a certain time limit, the help packet is sent again at an increased transmit power. This process is continued until the maximum transmit power is reached, or a valid packet has been received. If a node receives a “help” packet, it will increase its transmit power to match the value in the help packet and acknowledge the reception of the packet. If the sender of the help packet receives multiple acknowledgements, it will only select one acknowledgement and request that all other nodes reduce their power to a previous state. The purpose of this packet is to send an alert, receive an alert as well as acknowledge alerts.

A special “power_packet” is defined to perform a simple topology control algorithm. The diagram below shows the format of the 32-bit power packet, followed by the explanation of each of 8-bit attributes.

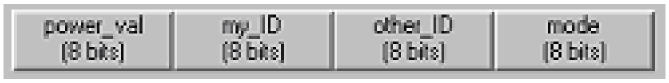

Figure 3: Power packet format

Power_val: this is an 8-bit floating-point value. The purpose of this attribute is to store the current transmitter power setting of the node that is requesting for help. Therefore, when a node receives a power packet and would like to acknowledge it, it will increase its transmitting power to match the value stored in this attribute. If its current transmit power is already larger or equal to this value, nothing is done.
My_ID: this is an 8-bit Object ID value. The purpose of this attribute is to store the ID of the node that is sending the power packet.
Other_ID: this is an 8-bit Object ID value. The purpose of this attribute is to store the ID of the node that initially sent out a request that has been received by the current node.
Mode: The “power_packet” has 3 different modes, as defined in the header block of the topology control module:

- **HELP**: The current node is requesting for other nodes to increase their transmit power. Any other nodes that receive this packet should respond accordingly by increasing their transmit power (if not already at maximum level) and then send out an acknowledgement.
- **ACK**: The current node acknowledges a help request by another node by increasing its transmit power and sending a “power_packet” in this mode.
- **NOH**: A node that receives multiple ACK from its original request for help will only select one acknowledgment (the first one that arrives). It will then send “no help needed” packets out to all nodes but the one that it accepted help from. This way, all other nodes can decrease their power level to the level before a help request was accepted.

Figure 4 below presents the process model of the topology control module. This proof-of-concept topology control algorithm is not designed as a robust or an energy efficient protocol. Instead, we provide a simple topology control algorithm that simply increase the transmission power level of each sensor node until every node can reach at least one other node. This algorithm clearly does not guarantee network connectivity, but demonstrates the distributive control of transmission power levels and its impact to the energy control module. Due to space limitation, the detailed discussion of the topology control algorithm is omitted in this paper.

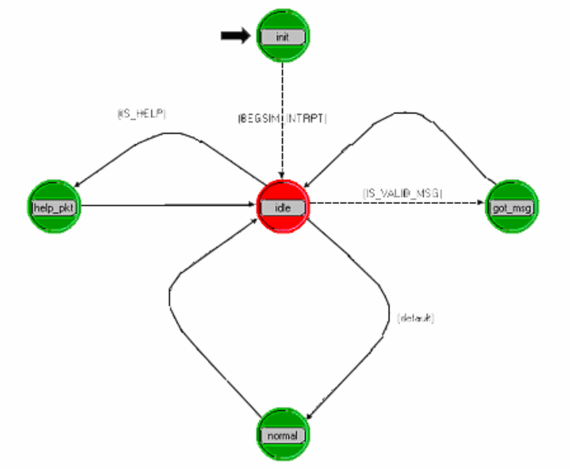

Figure 4: The process model for the topology control module.

5 Simulation Results

Various simulation results are presented in this section to validate the sensor network operation governed by the energy control module and the topology control module.

a. Validation of the energy control module

We first demonstrate the difference in energy consumption due to the differences in the traffic load. Consider the network topology shown in Figure 5, where 6 nodes were concentrated on one part of the layout and 3 other nodes were spread out in other areas. The simulation was run for 10 minutes and the remaining energy of the various nodes is plotted in Figure 6 and 7.

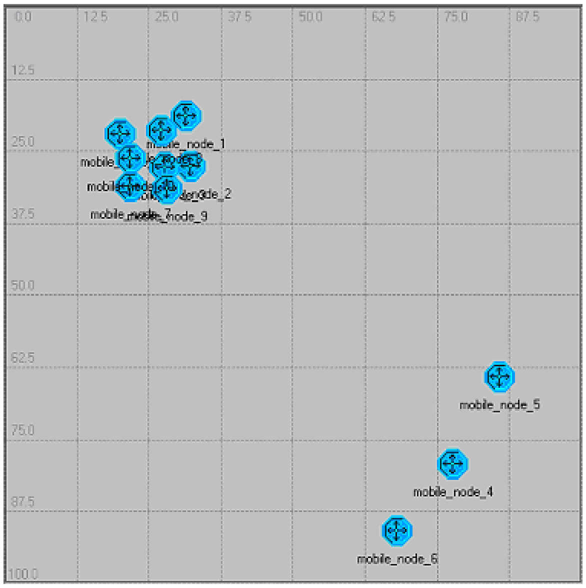

Figure 5: Network topology for simulation

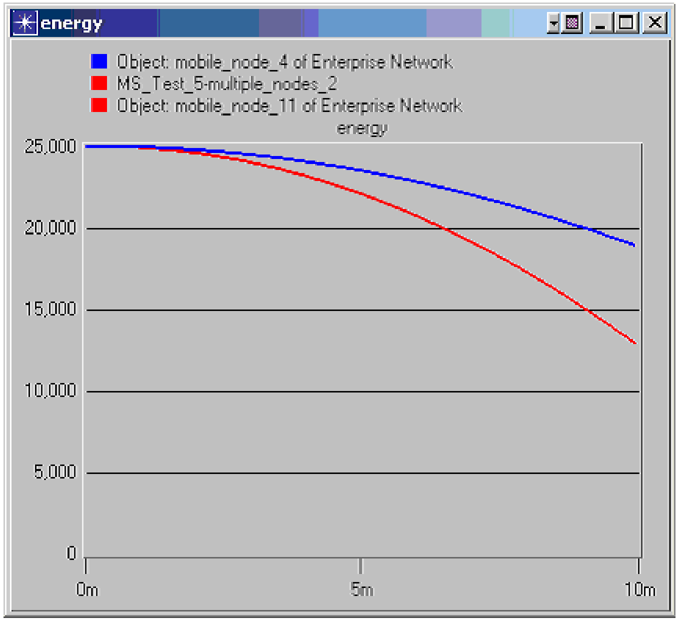

Figure 6: Residual energy for a node in the dense area and for a node in the sparse area

Since the density of the nodes situated in the upper left area of the scenario is higher, more packets are sent and received by these nodes, as compare to the other 3 nodes situated around this dense area. Due to this traffic difference, the energy goes down quicker for those in the dense area.

The next simulation demonstrates a node being disabled as it runs out of energy. The node's idle consume rate is increased from the default value of 0.05 Joules / second to 0.1 Joules / second to accentuate the node failure. Figure 7 shows the residual energy of the node and its transmission and receiving data rate. As can be seen, the node is disabled at around 2.9 minutes. At this moment, all data received and sent are stopped and the node does not respond to input by other nodes.

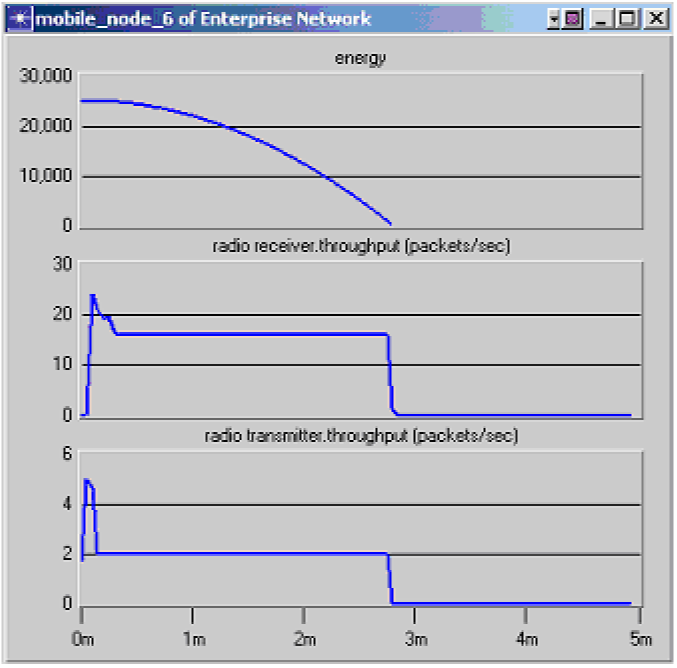

Figure 7: Disabling of mobile node due to lack of energy

When a node enters a sleep mode, its transmission and receiving of data is halted. The next simulation puts the node in a sleep state for the first 50 seconds of the simulation and awakens it in the next 50 seconds before putting it back in a sleep state 50 seconds later. The simulation graphs are show in Figure 9.

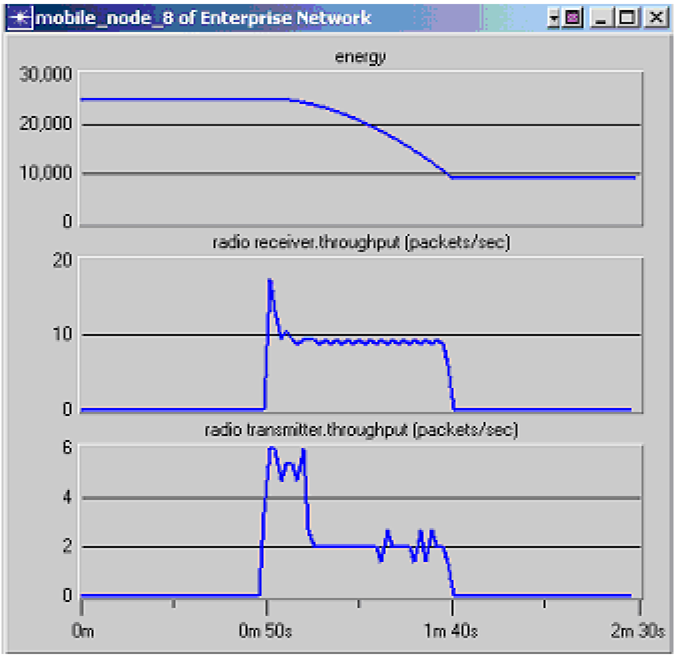

Figure 8: Effects of sleep mode on a mobile node

The idle consumption rate has been increased to 1 Joule / second in this simulation to accentuate the consumption that happens during normal mode. It can be seen clearly that during sleep modes (0s to 50s and 150s to 200s), no data is sent to the transmitter or received by the receiver. However, during normal operation, the radio receiver and transmitter are able to send and receive data as usual. During this normal operational time period, more energy is consumed because the idle consume rate and the packet transmit rate are larger than the sleep consume rate; thus, a huge drop in available energy can seen.

b. Validation of the topology control module

The transmission power of a standalone node, i.e., it has no neighbor it can ever reach is shown in Figure 9 when the topology control module is applied. Because of its stand-alone nature, this node will not receive any acknowledgement packets to respond to the power control packets. Therefore, the node will continuously increase its transmission power until the maximum allowed is reached.

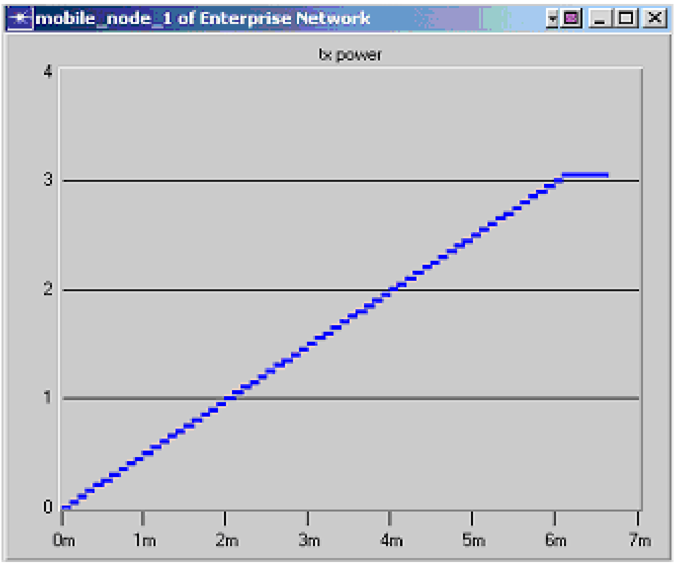

Figure 9: Transmission power of a standalone node

The next simulation is done with 2 nodes present in the project layout. To emphasize the role of "help packets" in establishing connections, the increment of Mobile node 0 has been changed to 0.01W while the increments of Mobile node 1 remains at 0.001W. The simulation was run for 400 seconds, and the results are shown in Figure 10.

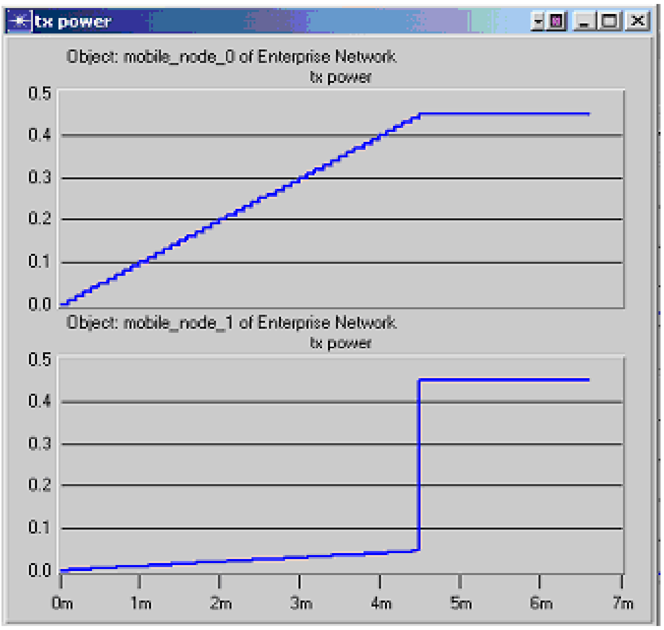

Figure 10: Transmission power of 2 neighboring nodes

As can be seen, the transmission power increments are larger for Mobile node 0 than for Mobile node 1. Both nodes keep sending out "help packets" until the packet from node 0 is received by node 1. This instructs node 1 to increase its transmission power to match that of node 0. The throughputs (packets/sec) of both receivers are shown in Figure 11. It shows clearly that both receivers begin to receive data only after the topology control module has established the proper communications.

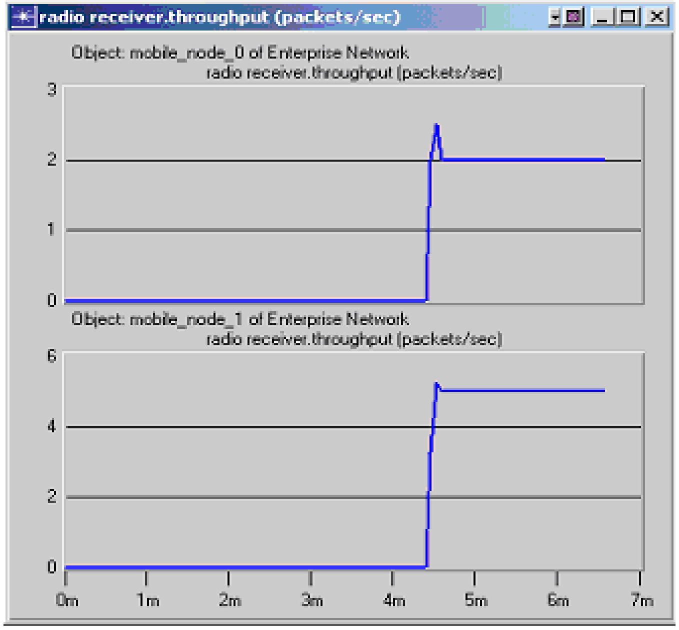

Figure 11: Receiver throughput of the two neighboring nodes

The final simulation is done with 3 nodes that are of unequal distances to each other to understand the effects of transmission power on their transmission range. The transmission increment of node 1 is 0.05W, 0.01W for node 2, and 0.001W for node 0. Figure 12 shows the change of the transmission power levels for the three nodes over time.

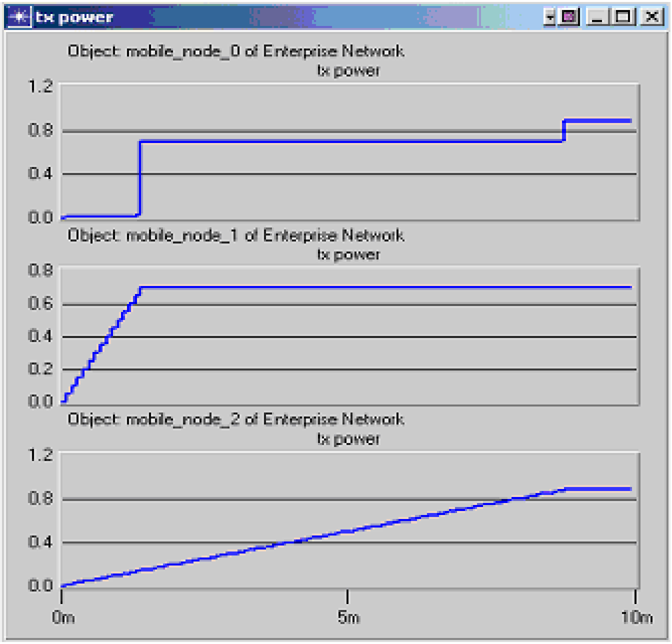

Figure 12: Transmission power of the three mobile nodes

As can be seen, node 0 has the smallest transmission power increments that almost look like a flat line from 0s to around 60s. At this time, it receives a help packet from node 1 and therefore increases its transmission power to a constant that is around 0.7W, as does node 1. Meanwhile, because of its distance, node 2 does not have enough transmit range until around 480s into the simulation whereby node 0 picks up a help packet from node 2. Therefore, node 0 increases its transmission range to match node 2 and this is round 1.0W.

Figure 13 below shows the receiver throughput (packets/sec) of node 0 and its transmission power. It can be seen that once connection with node 1 is established, the throughput increases to a constant of 0.7 packets/second. However, once connection is also established with node 2, then the throughput increases to 0.9 packets/second.

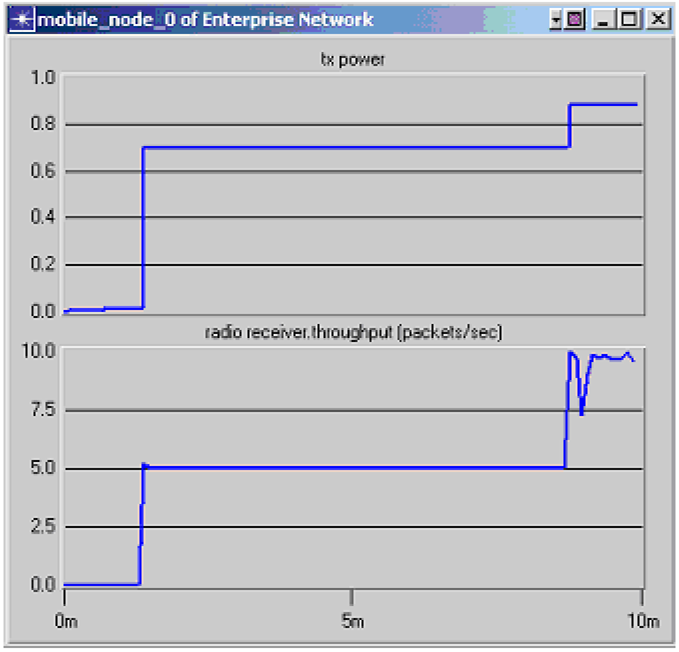

Figure 13: Transmission power and radio receiver throughput for node 0.

6 Conclusion

This work is the first step towards a modular simulation environment where the impact of network protocols on energy consumption can be examined. An energy module is developed in OPNET so that various wireless sensor operational modes can be invoked by network protocols. As energy runs out, the corresponding sensor node will be disabled to model the ad hoc nature of wireless sensor networks. A simple topology control protocol is also developed to test out the control of transmission energy and its impact to energy consumption over time. A new venue of network operation and modeling has been opened up. The development of this framework is expected to ease and further research on wireless ad hoc sensor network.

References

[1] Xinsheng Xia, “Ad Hoc Wireless Sensor Node of ABCP protocol,” OPNET user contributed models.

[2] V. Bahl, R. Wattenhofer, L. Li, and Y.M. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” pp. 1388–1397, Proceedings of IEEE INFOCOM, April, 2001.

[3] R. Ramanathan and R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” Proceedings of IEEE INFOCOM, March 2000.

[4] Yu Wang Xiang-Yang Li, Wen-Zhan Song, “Efficient topology control for wireless ad hoc networks with non-uniform transmission ranges,” ACM WINET, 2003.