Sensor Networks Modeling and Simulation in Ptolemy II

Philip J. Baldwin

Electrical Engineering Department, University of California in Berkeley
8/8/03
E-mail: pjb2e@virginia.edu

Abstract

With the use of micro sensors on the rise in technologies today a need for a better understanding of the interactions of such sensors is crucial. Many challenges arise when trying to model and simulate networks of these sensors (also called sensor networks). Sensors have various capabilities and constraints that provide this difficulty. A complete model of a sensor network is needed to improve upon its efficiency and design. Using a modeling and simulation tool called Ptolemy models of these networks can be better constructed. The research done in this paper will discuss the implementation and creation of a sensor node in Ptolemy. In the Ptolemy analyses of a sensor network, several factors affecting a network are realized such as battery life and signal range of the device. Along with modeling factors that can negatively affect the usefulness of the sensor, simulations of sensor nodes with extended sensing capabilities are also supported. Such capabilities include a seismic/sound detection and location algorithm along with data handling methods in the sensor network. Many applications are possible as we learn more about wireless communication in a sensor network environment.

1. Introduction

Wireless communication has become common place in everyday life. Hardly a day goes by without the use of some sort of wireless technology such as a cell phone or keyless entry system for a car. As CMOS technology continues to advance and circuits get smaller and smaller, it is becoming possible to place wireless circuitry in most electronic devices [5]. This advancement in technology has also brought about development of wireless micro sensors. These micro sensors have the ability to perform numerous sensing tasks from that of sound detection to seismic sensing. This makes it possible to collect data in remote areas or areas not suitable for human exploration. Networks of sensors or sensor networks have many uses in existing applications. Automobiles can use wireless micro sensor to monitor engine status, tire pressure, and fluid levels. Assembly lines could use these sensors to monitor process steps during production. In strategic situations micro sensors can be dropped from aircraft into hostile territory. These sensors can then be used to track moving targets, such as cars or people. Many other possibilities are available of which range from urban traffic monitoring to monitoring wildlife.

The variety of uses for sensor networks seems almost limitless. Because of this it is important that our understanding of the interactions of the sensor nodes is complete, therefore making it possible to design more efficient networks. Along with the importance of understanding how the sensor nodes work with each other is it also important we understand, first the reasons a sensor network can fail, and second the extended capabilities of a sensor network. In the design of the sensor network model the aforementioned important factors of a sensor network will be kept as tasks that have to be completed before an accurate simulation can be achieved. These tasks are summarized as the following: 1) *Understanding the interactions in the sensor network-* Data management through wireless communications and wireless channel and port configurations. 2) *Sensor network failure* – The affect of power lost, signal strength, external influences, and strained resources on a sensor node in multiple environments. 3) *Extended capabilities of a sensor network* – The addition of processing and sensing features such as seismic, sound, IR, motion, and increased data processing; along with sound and motion tracking algorithms. Upon completion of these three main modeling tasks a proper simulation of a sensor network can be developed.

- *Understanding the interaction in the sensor network.* Before a model of a network is possible you must know how data is passed around the network. When messages are first broadcasted out to the network. The decision on which sensor node gets to broadcast its message first varies depending on the type of wireless protocol. The Bluetooth wireless protocol requires that no single node is assigned a permanent master/slave role. During runtime the master/slave roles switch between broadcasting and receiving nodes. Other technologies such as 802.11b have similar operation protocols. Another widely used setup is that of one root sensor node in a network of branch or leaf nodes. The root node sends out the initial data and the branch nodes pass on that information to other nodes in the network. As a starting place in our model a protocol similar to that of Bluetooth was implemented. Since there is no initial assignment of master/slave nodes, the node that gets to broadcast first is simply the node with the lowest ID or index assigned by the designer. The sensor nodes in the network make broadcast then check to see if a message was received at the input port. Improving upon these methods is one of the main ways to increase efficiency.

- *Sensor Network Failure* – Sensors nodes have limited resources and cannot operate for an infinite amount of time[5]. Predicting sensor node failures is important in creating a network model. Some factors that can cause a network failure are battery power, geographical changes that can cause sensor nodes to go out of range, signal strength, and processing problems. These issues will be addressed later in the paper.
- *Extended capabilities of a sensor network* – Sensor nodes can be extended with many sensing capabilities to improve the functionality of a sensor network. Such sensing possibilities include sound and motion detection. This application could be used to track vehicles in remote locations [5]. Another sensing possibility is that of seismic detection. Predicting earthquakes have long been an engineering problem. Using seismic sensor nodes it would be possible to gain a better understanding of what happens when earthquakes occur or if there's any warning seismically that an earthquake may occur. Uses of sensor networks are constantly growing. These sensor networks are key in information gathering and data processing. Many applications can be improved by the use of micro sensors.

2. Heterogeneous Modeling and Design

Ptolemy II

"The Ptolemy project studies heterogeneous modeling, simulation, and design of concurrent systems. The focus is on embedded systems [1], particularly those that mix technologies, including for example analog and digital electronics, hardware and software, and electronics and mechanical devices. The focus is also on systems that are complex in the sense that they mix widely different operations, such as signal processing, feedback control, sequential decision making, and user interfaces [2]". The Ptolemy II project studies heterogeneous modeling, simulation, and design of concurrent systems. The Ptolemy II software has the ability to model complex systems through a simple actor/component based user interface. One of the key principles of the Ptolemy project, as sited on Ptolemy objectives website, is the use of multiple models of computation in a hierarchical heterogeneous design environment [3]. The ability to model more than homogeneous structures makes Ptolemy a very versatile modeling tool. This is of particular interest in the sensor domain. There are many different factors at play in the operation of a sensor network. Being able to capture all this factors in a model is made easier by using Ptolemy.

3. The Sensor Network Model

Networks of sensors prove invaluable for information gathering applications. Sensor networks have the ability to monitor multiple areas geographically and process the data in the network. Designing the infrastructure of these sensor networks can be very challenging.

As previously discussed the three main tasks (Understanding the interactions of the sensor network, Sensor network failure, and extending the capabilities of the sensor network) are the main goals of the model to be designed in Ptolemy. The starting point of the design was the creation of a process flow or a step by step design technique for each component to be included in the sensor network model in Ptolemy. The process flow is as follows:

Process goals -- To generalize a sensor actor, sensor director, and a wireless IO port to be used with multiple sensor network designs in Ptolemy.

Sensor Node–

- Needs to be a general sensor that uses a wireless IO port and a sensor director.
- This sensor well be used as a node in a wireless network
- No hard coding of algorithms should be placed inside
- Sub classes can be models with own specifications
  - Sensor Node – Process Flow
    1. Create wireless port parameter for use
    2. hide the parameter
    3. Setup up attributes
       - Signal radius
       - Icon
       - Circle size
       - Antenna

- Color

4. Fire
   - Broadcast to other nodes in reach
   - Check battery power left
   - Adjust Signal Radius
   - Information Check
5. initialize
   - create circle
   - color
   - get location of node

Wireless IO Port –

- This needs to be a general port used by anything in the sensor domain.
- Should only be a port for transfer of information
- This port is the default sensor domain port, just as the generic port is for a regular actor
- Should have code of the Sensor director.
- This will be the port that hands any type of wireless communication in Ptolemy.

Sensor Director

- Needs to control in a general anything having to do with wireless transfer of information
- Uses wireless IO and wireless receiver
- Keeps track of what's in reach of the sensors, and places the information at the wireless IO port
  - Sensor Director – Process Flow
    1. Index all nodes in network
    2. Sequentially determine all nodes in range
    3. Fire nodes in broadcast range
    4. Put info(Token) at the Port(wireless) of each sensor within range
    5. Is reachable
       - Return if another node passed in by the function is in broadcast range of the node
       - Calculate distance
       - Sensor Director keeps track of all nodes in reach, then fire the node to broadcast to all the nodes in range.

Observations

Color:

- Use color to indicate when broadcasting or receiving
- Color is set up with 4 variables from a 0 to 1 scale. {Red, Green, Blue, Alpha}
- Alpha is the transparency (1 is opaque)

Using a previous sensor network model created by graduate students Cheng Tien Ee, Sanjeev, and N. Vinay Krishnan at the University of California, Berkeley I was able to create the basic infrastructure for a new, more general sensor network. The new infrastructure contains four main components: the sensor node, the wireless IO port, the sensor director and the wireless channels.

The Sensor Node

Was the first to be modified and updated for the new sensor network model. Following a process flow created early on, the sensor node had to be made general. This meant that no network optimization algorithms could be hard coded in the sensor. Optimization of the network would be added in more complex models. The sensor node is this model was to be a basic, simple sensor. This simple sensor would create a wireless IO port and check if it received a message at its wireless input port. If no message is present the node then broadcasts a message to all nodes in reach (directed by the sensor director). In Ptolemy it is possible to make a sensor node out of a graphical component interface. The sensor nodes characteristics can be attributed to the model using composite actors. Another option is to create the sensor node object in java. This was the choice for the sensor node implementation. *See figure 3.1 for java code of simple sensor shell and figure 3.2 for graphical icon.* At this stage in the creation of the sensor network model, there exist sensor nodes that have the capability to broadcast and receive messages from other nodes. What is now needed is a specialized IO port to handle the broadcasting and receiving of these wireless messages in the network.

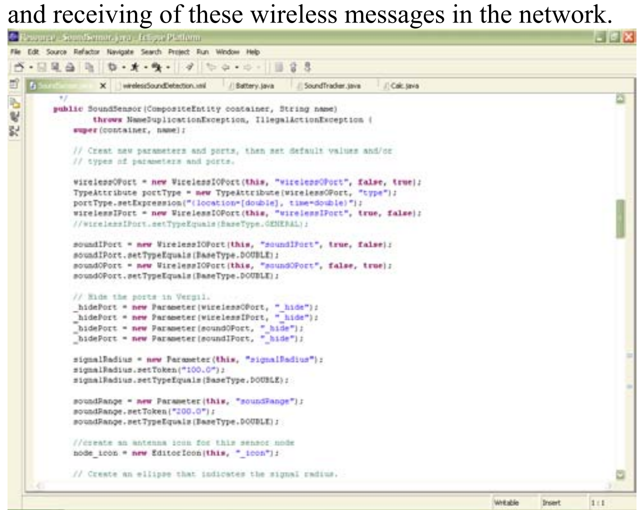


Figure 3.1: Java code for a Sensor Node

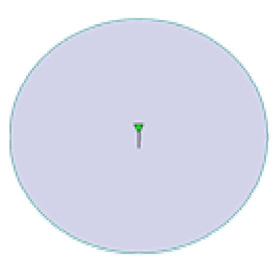


Figure 3.2: Sensor Node graphical Icon

The Wireless IO Port

Is a type of IO port in Ptolemy that is specialized for sending and receiving wireless messages. Two wireless IO ports are created for each instance of a sensor node. One serves as the input port for receiving a wireless message and the other serves to send a wireless message. The wireless IO port sends/receive messages from a channel that matches the name of the port. For instance, "wirelessIPort" will receive messages from the wirelessChannel. The "wirelessOPort" will send messages to the wirelessChannel. A possible extension to this implementation, which will be covered later, is the creation of multiple channels in the model. To add a sensing capability to a node all that needs to be done is had a channel for that sensing process (i.e. Sound Channel..etc). At this stage in the creation of the sensor net model, the sensor nodes have ports for sending and receiving messages along with the channel for the wireless passed around the network. All that is needed now is a component to coordinate and direct the firing of the sensor nodes and move the messages around the network.

The Sensor Director

Directs all actions and processing in the sensor network model. The sensor director is based on a discrete event director in Ptolemy. The modifications of the sensor director were done by graduate mentor Xiaojun Liu. The new director is called by the wireless channel to determine which nodes can hear the broadcast. The *isReachable* function calculates the distance between nodes and determines which nodes are in reach of the sensor node parameter *signalRadius*. This parameter determines how far a sensor nodes message is heard or broadcasted. Once message is retrieved from the input port it is ready to receive another wireless message. The sensor director will invoke a sensor node when it receives a input message from a channel. The sensor node is then allowed to schedule a new command to run itself at a future time. A possible extension to this model is a way for the sensor director to randomize the nodes in the simulation window. This could serve as a representation of a deployment of sensor nodes from an aircraft or a moving vehicle. The randomized nodes method has been partially implemented in the new sensor director. At this point the simple sensor network model is complete with limited capabilities. The sensor nodes can all broadcast a message which will be received by only the nodes that or with in the broadcasting nodes signal range. Once a sensor node receives a message it can then decide what to do with it. In this first simple model all the nodes broadcast there location and time.

Communication Channels

Determine how messages are transferred among the sensor nodes. Along with that it determines what nodes are reachable, the communication delay, and reliability. The current implementation has a wireless channel which has no delay; the range is determined by the senders signal range parameter, and reliable. The second channel implemented is the sound channel, where the delay is the propagation speed of the sound, and because of this delay a sound message will reach different receivers at different times. The communication channels are the back bone of the sensor network system; they're responsible for all messages that are passed around the network.

4. Creating a more accurate Sensor Network model

The Battery Model

Creating an accurate battery model is important when thinking about the effect the battery can have on a sensor network. There are several factors to consider when trying to model a battery. The main factors to consider are battery capacity, voltage and energy. While in use, these three variables are constantly changing. As the battery capacity drops, voltage levels can only be sustained for a finite amount of time. Once the battery is below a certain threshold voltage there is not enough power to for the device using the battery [4]. In the current implementation the sensor node creates a battery object. The battery is notified each time the Sensor node broadcasts. After each broadcast the battery recalculates its battery life and voltage. When the sensor node detects that the battery is reaching its threshold hold voltage it will attempt to increase battery life by reducing the load on the battery. The sensor node is able to reduce the load by reducing its signal radius. *See Figure 4.1.* Accurate predictions of batteries behaviors are very challenging even to battery manufactures. This battery model is a start in predicting the behavior of a sensor network with nodes that have finite amounts of battery power.

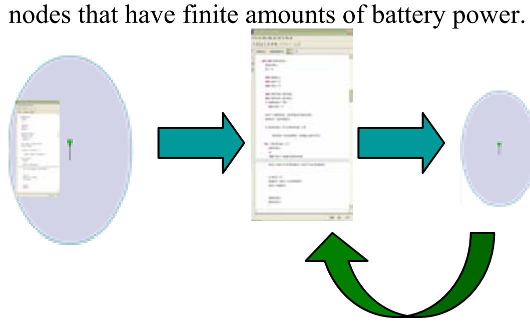


Figure 4.1

5. Sensor Network Simulation

After the completion of the main components of the Sensor network an application to test the model can be created. Since a sound channel has been implemented a sound source tracking application was created *see figure 5.2*.

The sensor nodes were modified to be sound sensors. This was accomplished by creating sound input port in the sensor nodes. Since the type of channel used depends on the port name this port will be controlled by the sound channel. The sound sensors were also given wireless output ports to talk to each other and to other wireless devices.

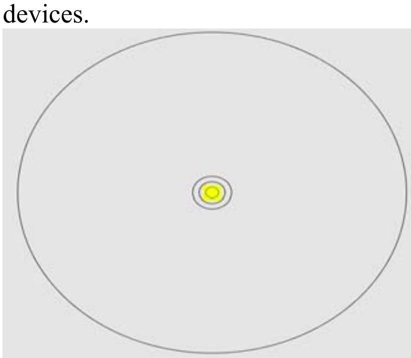


Figure 5.1: Sound Source

Next a Sound source (*see figure 5.1)* was created to broadcast a sound signal using a sound output port. The sound source moves around the simulation broadcasting a sound message. The sound source has the ability to move around the window by setting is location during runtime. New locations are updated to a circular path around the sound sensors.

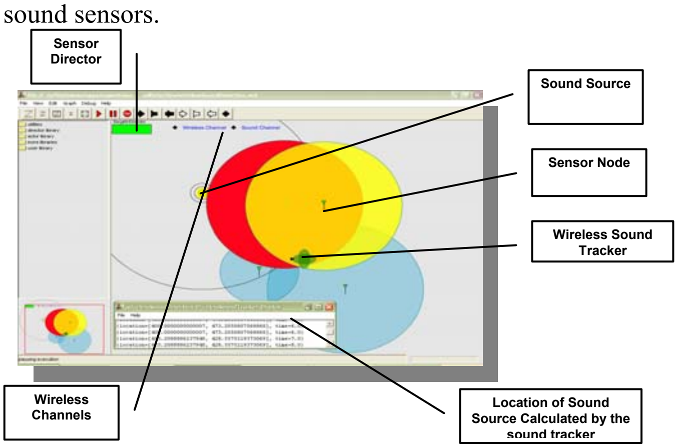


Figure 5.2: Sensor Network sound tracking application

The next part to the creation of this application was the Wireless Sound Tracker actor *see figure 5.3*. This actor uses a java object called sound tracker which in turn uses a triangulation algorithm to determine the location of the source based on information received from the sound sensors. The Wireless Sound Tracker receives information from the sound sensors via the wireless channel. The Wireless Sound Tracker has one wireless input port that receives the data from the sound sensor's wireless output port. The sound tracker actor is able to determine the location of the source based on the location of the sensor when it received the sound message. The sound tracker needs to have the values from at least three nodes in order to apply the triangulation algorithm.

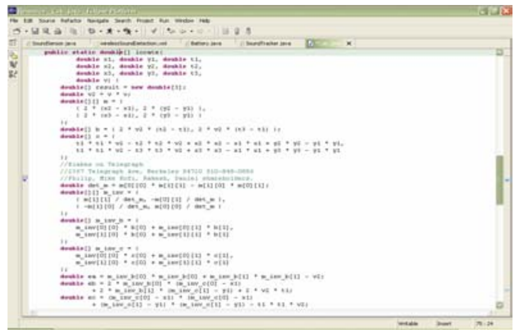


Figure 5.3: Calculation method to determine the location of the source.

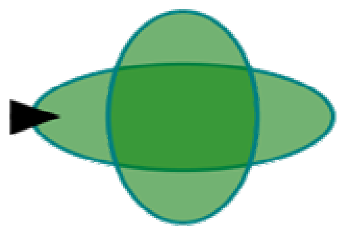
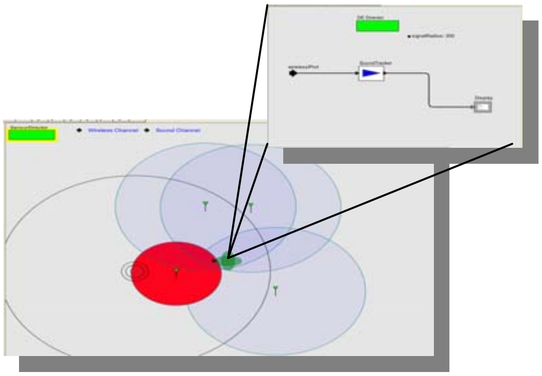


Figure 5.4: Wireless Sound Tracker

Run Time Simulation

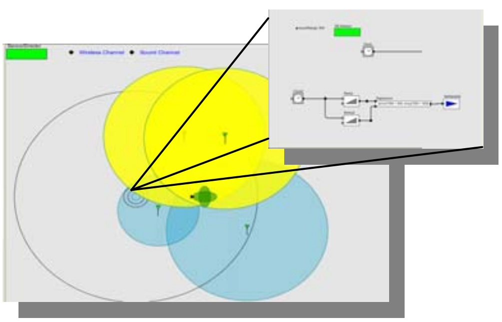
1) The Wireless Sound Tracker receives wireless broadcast from sensor nodes. These broadcasts are sent to the Sound Tracker component that uses a triangulation algorithm to locate the source.

2) The sound produced by the sound source is heard first by the smaller sensor node. This is indicated by the node changing its color to red. When the sound source makes a broadcast the nodes closest will here it first. The sound channel determines at what time each node hears the sound based on there location in the model and the propagation speed of sound.

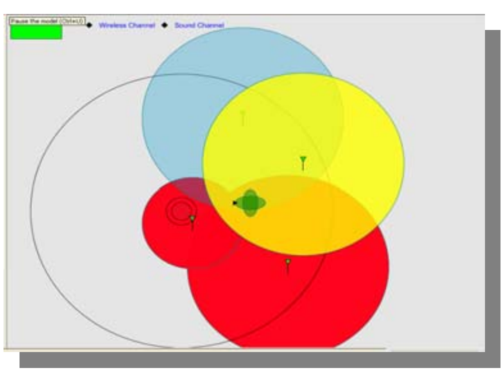


3) The Sound Source sends out a wireless sound message at a pre-specified clock period. The sound message is carried through to the nodes via the Sound Channel.

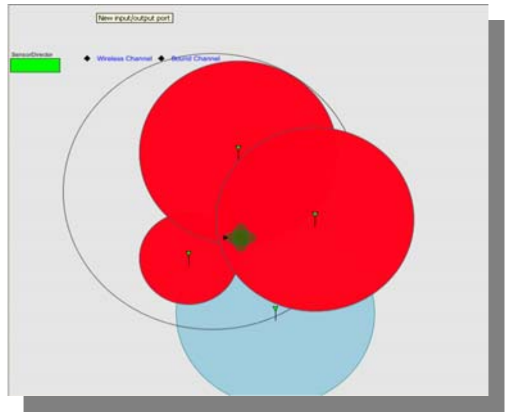
4) Two of the nodes in this simulation have both received a wireless message. This is indicated by the yellow color of the circle. The two other nodes are broadcasting at this time period. This indicated by the blue circle color). The sending and receiving of wireless message are facilitated by the Wireless Channel.



5) In the next step, two of the nodes have heard the broadcast of the sound source while another node is receiving a wireless message and yet another node is broadcasting.



6) In order for the Wireless Sound Tracker to calculate the position of the Sound Source it must receive wireless messages from at least three nodes. Once three sensors hear the sound (at least three red circles), they must then broadcast a wireless message to the Wireless Sound Tracker.



6. Conclusion and Future work

Since the model was created to be expandable, addition sensing capabilities can be added by simply creating a new wireless channel for a particular detection task just as a sound channel was created for the sound sensing capability. The application above is an example of how by making simple but expandable sensor network models it is possible for complex sensing task to be achieved. Although this is just a starting point to a complete sensor network model, this simple sensor network model acts as a stepping stone for more complex designs. Future models could include but not limited to the following:
-- Seismic Activity
-- Heat Detection
-- Motion Detection
-- Target Tracking.

References

[1] Edward A. Lee, "What's Ahead for Embedded Software?," IEEE Computer, September 2000, pp. 18-26.

[2]Edward A. Lee Technical Memorandum UCB/ERL M01/11 March 6, 2001.

[3] The Ptolemy Project. http://ptolemy.eecs.berkeley.edu/

[4] Panasonic Battery models http://www.panasonic.com/industrial/battery/oem/images/pdf/Panasonic_Alkaline_Hdbk_02-03_v1.pdf

[5] Sung Park, Andreas Savvides, and Mani B. Srivastava, "SensorSim: A Simulation Framework for Sensor Networks

Acknowledgements:

Professor Edward A. Lee, for all the advice and ideas for my research

CHESS faculty and staff

Special Thanks to my parents Tony and Naomi Baldwin for letting me travel 2,000 miles away to do research.

Acknowledgements:

Xiaojun Liu – Staying late to help out and being a great mentor

Blakes att: Riya Kuo

Superb- IT