Building and adapting a Multihop Wireless Sensor Network

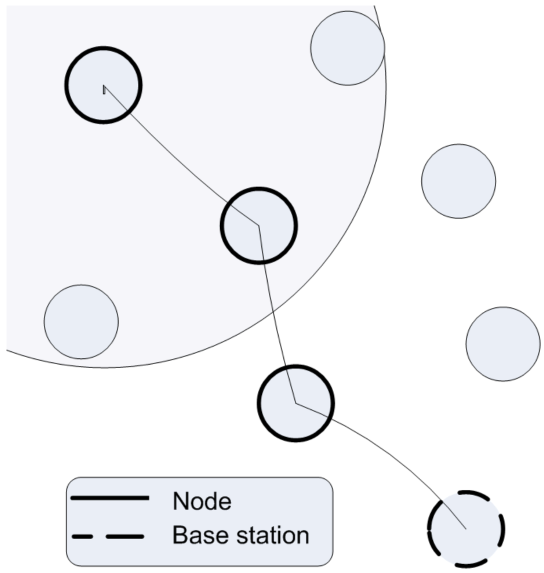
Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen.

Abstract—The purpose of this thesis is to investigate the existing multihop networks in the TinyOS environment and to implement them in an existing network. Furthermore, we'll add to cape ability to utilise multiple sensors on the sensorboard and transmit the sensordata over the network. Finally, we'll adapt the protocol to include some basic routing functionality. Specifically we will make a lowest hop protocol.

Index Terms—WSN, Wireless Sensor Network, TinyOS, telosb, multihop, sensor nodes.

1 Introduction

A Wireless Sensor Network is a wireless network that consists of many small, autonomous nodes equipped with sensors and a radio. The main purpose of these devices is to monitor the environment and to transmit the data to a base station. This base station is usually a more powerful computer that aggregates and analyzes the data. The nodes in these networks are battery powered and should be able to last a long time on a single charge. Therefore, each component should use as little power as possible. In many applications these nodes have to monitor a large area, so the radio signal has to travel a long distance. Because the power our radio can transmit is limited, so is the distance the signal can reach. This would make the nodes insufficient for many applications, such as habitat monitoring, agricultural applications... A solution is to use a multihop routing protocol. It instructs the node to send its data to its nearest node, which will then relay the data one step further closer to the base station and so on. Eventually the data will reach the base station, with minimal power consumption. We studied the Collection Tree Protocol which is the standard multihop protocol available in the TinyOS 2.0x distribution. We implemented this protocol in a dynamic event positioning application. Section 2 briefly describes CTP, section 3 describes our work.



(a)

Fig. 1. A multihop network

2 Background

CTP is a protocol that uses a virtual tree abstraction to collect transmitted data in a multihop routing network. It provides best-effort anycast datagram communication to one of the collection roots in a network. Collecting data at a base station is common in a sensor network application. The collection of data is made possible by creating 1 or more collection trees, each with the root situated at the base station. When a node has data which needs to be collected, it transmits the data up the tree and it also forwards the collection data which other nodes send to it. When a network has multiple base stations that are configured as a root, then it has a forest of trees. The protocol will make a reasonable effort to deliver the message to at least one of the roots in the network. There are however no guarantees of delivery, and there can be duplicates delivered to one or more roots. There is also no ordering guarantee.

• *P. De Cauwer, T. Van Overtveldt, J. Doggen are with the University College of Antwerp, Paardenmarkt 92, 2000 Antwerp, Belgium*
E-mail: see http://www.e-lab.be

3 Main Goals

The research contained 3 goals. The first goal was building a multihop routing network. The second goal was implementing multihop in a project which uses the Received Signal Strength Indication (RSSI) to indicate the position of a node in a wireless network. The third goal was integrating our work with that of our Master students[1]. Therefore, the program needs to be compatible with the Octopus GUI[**?**].

4 Method

All of our work was done with Telos Rev B[2] motes, which have the following specifications:

- TI MSP430 microcontroller with 10kB RAM
- IEEE 802.15.4 compliant radio
- Integrated temperature, light, humidity and voltage sensor
- TinyOS 2.X compatible
- Programmable via USB interface
- Integrated antenna

Except for the sensors which are specific to our platform, our work should be portable to other platforms.

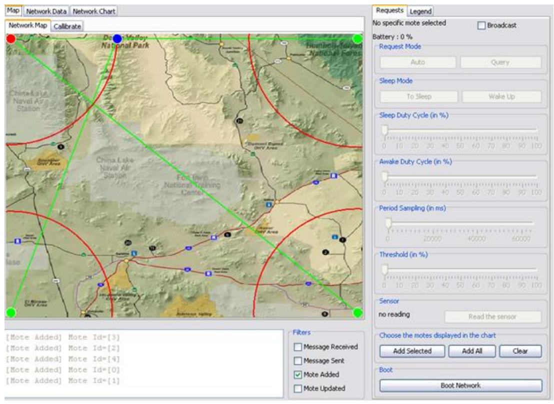
We build a network with 3 nodes: 1 acts as the root node (RN) and is connected to a computer, the second node (N1) is in range off the root and collects data. The third node (N2) is out of range of the root but still in range of N1. Each node collects data and sends it to the root node. The data from N2 is relayed by N1. We also expanded this application to make it work with multiple sensors.

For our second goal we wrote 2 applications. One node acts as an emergency beacon. When an event occurs, such as an extremely high temperature, the node starts sending an emergency broadcast. The second application is installed on another node. When it hears the emergency broadcast, it will start to measure the RSSI of the packet. The node is then instructed to travel in an arbitrary direction towards the alarming node. When it notices that the RSSI improved, the same direction is used. If the RSSI got worse, it will use a different direction. We calculated the RSSI by the method described in [3]Eventually, when the destination is reached, a user defined action will be performed. For example, a picture could be taken, so that the user can evaluate the situation.

For our final goal we made our applications compatible with the Octopus GUI. The octopus GUI is a GUI which visualizes all the nodes in a wireless network based on the received packages. The packages are of the form message_t which has a payload of the type struct. The struct contains many variable that the GUI needs to localize all the nodes. The frame format is as follows:

Payload structure of Packet		
DataType	*Variable*	*Description*
nx_am_addr_t	moteId	Node ID of the sending node
nx_uint16_t	count	Sample count
nx_uint16_t	quality	Quality of the link
nx_am_addr_t	parentID	The parent ID of the node
nx_uint8_t	reply	Field used by the root node to reply on the received messages
nx_uint16_t	Voltreading	Reading of the voltage sensor
nx_uint16_t	Lightreading	Reading of the light sensor
nx_uint16_t	Tempreading	Reading of the temperature sensor
nx_uint16_t	Humidityreading	Reading of the humidity sensor
nx_uint16_t	RSSI	Value of the RSSI with reference to a certain anchor node in the wireless network

Figure 2 shows the Octopus GUI. The network that is displayed consists of several motes, each with a different function. We have anchornodes which monitor the environment and serve as reference point for the positioning. Then we have the mobile node, it reacts to an alarming signal received from a anchor node and moves in that direction to investigate the problem. Finally, the base station, receives the data from each node and sends it to our GUI.



(a)

Fig. 2. The Octopus GUI

5 Conclusion

We can conclude that CTP is functioning well and that we were able to build an application on top of this protocol. The RSSI readings are very unstable, even in the open air, with line-of-sight. We can conclude that the tracking of a node with the use of anchor nodes is imprecise at small distances. Expect an accuracy of 3 meters in outdoor applications. We have succeeded in integrating RSSI readings with the Octopus GUI. However much more work needs to be done before we have a usefull application.

6 Future Work

An algorithm to process the RSSI data and calculate a real-time position still needs to be done properly. This needs to be written in nesC so that the mobile mote can be independant from the base station. A second proposal is interfacing the mobile node with a robot so that it can move to the site of the event.

References

[1] N. Verbaendert and D. Henderickx, "emote: A wireless sensor network monitoring tool with node localization using rssi," p. 7, 2008.
[2] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," p. 6, 2004.
[3] J.-P. G. L. D. S. Geoffrey Ottoy, Anneleen Van Nieuwenhuyse, "Indoor localisation techniques: Comparison between the use of rss and toa," vol. Vol. 26, No. 10, p. 4, 2007.

Student 1 Peter De Cauwer is currently working on his thesis for his bachelor degree in Applied Engineering: Electronics-ICT at the University College of Antwerp

Student 2 Tim Van Overtveldt is currently working on his thesis for his bachelor degree in Applied Engineering: Electronics-ICT at the University College of Antwerp

If you wish to cite this paper, please use the following code: ”Peter De Cauwer, Tim Van Overtveldt and Jeroen Doggen, Building and adapting a Multihop Wireless Sensor Network, Bachelor’s Thesis, University College of Antwerpen, Belgium, June 2008”