Program 3 {February 22, 2013} 48 Points

A Network of TelosB Sub-Networks

Due: Thursday, February 28, 2013 at 10 a.m.

Introduction

The goal of this project is to have each team manage communications between three motes in their own sub-network and also interact with a base station and a target node. Optimistically, the bigger objective is to run an experiment in class on February 28th that includes all the sub-networks in the classroom.

Assignment

Figure 1 is a characterization of the setup that your team would use to test your implementation where the nodes labeled **Base Station** and **Target Node** will be implemented by **BS/MS team** and the other three nodes represent your team's three-node sub-network. For this assignment, your subnetID is your team number and your **new** nodeID ranges are **defined on the Programming Teams' pdf file**. For sub-net communication use the assigned, dedicated frequency defined in the same pdf file. The initial action for your three motes is to turn off their LEDs. Once they receive *BeaconMsg*'s, the motes should turn on their Blue LEDs to indicate they are connected to the **Base Station**. The **Base Station** and the **Target Node** will only use the broadcast channel. The broadcast channel is **DEFAULT_FREQ_CHANNEL** (namely, 26).

Your subnet test experiment is as follows. The **Target Node** sends out a *TargetMsg* every **TARGETPERIOD** (with a constant transmit power) to the **TOS_BCAST_ADDR** over the broadcast channel. All your sub-network motes should detect these messages because all sub-net motes are listening on their private channel and the broadcast channel and use the messages to 'find' the **Target Node.** Specifically, your motes need to communicate among each other on their private channel to determine which of the three motes is closest to the **Target Node** by using received signal strength (RSSI). The node in your sub-net closest to the target will be identified as the **Near Node (**the near node needs to be unique. Use smaller nodeID as a tie-breaker). Once this determination has been made, the **Near Node** turns on its Red LED. The **Middle Node** simply leaves its Blue LED on and the **Far Node** turns on its Green LED. Since the **Target node** is mobile, the identity of the **Near Node,** the **Middle Node** and the **Far Node** will change over time.

The **Near Node** must communicate its nodeID occasionally to the **Base Station** over the broadcast channel via a *ReportMsg*. This occurs in two situations:

1. A *ReportMsg* is required every time the identity of the **Near Node** is switched to another mote in a sub-net.
2. The **Base Station** may inject a REQUEST for a *ReportMsg* on the broadcast channel.

The *ReportMsg* should be sent to the **Base Station** after either of the two possible events. The *ReportMsg* contains a TimeStamp which is the time the message was generated according to the **Base Station** clock. Since each node keeps its own local time, you might need a time synchronization protocol **{Note – given the shortage of time, this synchronization protocol is beyond the scope of program 3}**. The request field of the *ReportMsg* is explained below.

The **Base Station** sends out a *BeaconMsg* every **BEACONPERIOD** over the broadcast channel using the **TOS_BCAST_ADDR**. Simple beacons advertise the presence of the **Base Station** and its local clock. Complex beacon messages additionally serve as a REQUEST for a *ReportMsg* from an individual sub-network. The request field in the *BeaconMsg* identifies by subnetID the specific sub-net to whom the request is being sent. Hence, the resulting *ReportMsg* must return the value of the request field.

BeaconMsg's are also employed to evaluate connectivity. The objective is to have every mote be able to directly communicate with the **Base Station**. If a node does not hear the Base Station for five **BEACONPERIODs**, it is assumed disconnected from the **Base Station** and should indicate this by turning its Blue LED off. Since all motes start with their Blue LED's off, once a mote successfully receives a *BeaconMsg* of either type, it should turn its Blue LED on. Hence Blue LED on is visible signal that the mote is connected to the **Base Station**.

You will have to test your design against the working implementation of the **Base Station** and **Target Node**. Thus the actual value of **BEACONPERIOD** and **TARGETPERIOD** might change and should be used as constants in your design.

You are permitted to and probably should define new message types. When creating new active message IDs for your internal subnet, use numbers that indicate your subnetID in the first digit (e..g, if you are subnet 3, your AM IDs could be 31, 32, 33, …) Place new messages and new definitions in a new.h file. **MsgProgram3.h** will be provided by the TA to identify the other message types (*TargetMsg*, *ReportMsg* and *BeaconMsg*).

To control the radio settings, you must use the RadioControl interfaces as in Program 2. **Once the Base Station and Target Node have been implemented, they cannot be fully tested until one team completes its sub-net functionality. Note, the 'drop-dead' deadline for this assignment is the beginning of class on Monday, February 28th.**

Deliverables

Your team will need to provide a concise report detailing your design decisions. This should explain protocols chosen, tradeoffs considered and rationale for your key decisions. As with the previous programs, your team needs to turn in a tarred/zipped file that contains source code, make files and a README file. The README is critically important for receiving partial credit on this assignment and needs the current functionality of your motes at the time of submission.

While we will attempt a live demonstration in class on February 28th, each team may need to arrange a private demo slot with the TA **before the term ends** to go over your implementation and to return all your motes.

Figure 1 - Group Experimental Set Up