

# Lab Report – Radio Communication

Stuart Olsen  
so1132

October 2, 2014

## Introduction

The task of this lab was to set up two motes to communicate via radio. The “sender” mote sent out a repeating sequence of packets encoding numbers to be displayed on the “receiver” mote’s LEDs. The receiver mote was to be configured to send acknowledgment of received packets. The sender mote was to be configured to require receipt of this acknowledgment before sending the next packet in the sequence.

I learned how to use the TinyOS documentation and how to configure the AM radio send/receive components to use packet acknowledgment.

## Implementation

The sender was configured at boot (event `Boot.booted`) to require acknowledgment for the sent packets using `PacketAcknowledgements.requestAck`.

The code for sending packets was then modified only to continue to the next packet in the sequence upon receipt of acknowledgment. This was done by making the data increment conditional on the result of `PacketAcknowledgements.wasAked`.

The components used by the sender and the way in which they were wired is as follows:

```
implementation {
  components MainC, LedsC;
  components new TimerMilliC() as TimerC;
  components ActiveMessageC as AMsgC;
  components new AMSenderC(RADIOID);
  components SenderC as AppC;

  AppC.Boot -> MainC.Boot;
  AppC.Leds -> LedsC.Leds;
  AppC.Timer -> TimerC.Timer;
  AppC.Packet -> AMSenderC.Packet;
  AppC.AMSend -> AMSenderC.AMSend;
  AppC.SplitControl -> AMsgC.SplitControl;
  AppC.PacketAcknowledgements -> AMSenderC.Acks;
}
```

The interfaces used were:

```
module SenderC {  
  uses interface Boot;  
  uses interface Leds;  
  uses interface Timer<TMilli>;  
  uses interface Packet;  
  uses interface AMSend;  
  uses interface SplitControl;  
  uses interface PacketAcknowledgements;  
}
```

The receiver code did not require any modification to implement acknowledgment.

## Experiment

To test communication failure, the receiver mote was moved to the opposite side of the room from the sender mote. Occasionally, instead of incrementing the number displayed on the LEDs in step, the receiver would continue displaying the same number for multiple steps, then change to the number which would be shown if it had operated correctly; that is, the receiver would occasionally skip numbers.

After configuring packet acknowledgment, the receiver exhibited nearly the same behaviour, with one exception: when it resumed the sequence, it picked up on the next number in the sequence after the one on which it had stopped. This is because without receiving the next packet, the receiver would not update the LEDs and would not send an acknowledgment to the sender. Without the acknowledgment, the sender would continue to broadcast the same packet until the receiver acknowledged receiving it, and so the receiver could not skip displaying any number in the sequence.