







Nate Olderman & Brandon Roberts

3) Threads are used in this application whenever a process needs to occur simultaneously to the main process. This means that there are two threads, the sender and the receiver. The sender will be waiting for data and source address from `send()` in `LinkLayer`, adding on the necessary frame bits, and sending it through the network. The receiver will be waiting for and receiving packets from the network and giving it to `LinkLayer` for use in `its recv()`.

4) `send()`:

- Compares the length of the given `data` buffer to the given `len`
 - If `len` is greater than `data`'s length, it will give all the bytes in `data` to the sender thread to send in addition to the destination address and the RF layer
 - If `data`'s length is greater than `len`, it will give the first `len` bytes of `data` to the sender thread to send in addition to the destination address and the RF layer
- `send()` will then return with the amount of bytes that it passed to the sender thread
- The sender thread, once given the data and destination address, will create a packet object to build the frame around the data.
 - this includes adding control, destination address, source address, and CRC as well as the necessary bit manipulation before sending
 - Then the thread will go through the process drawn out on the finite state diagram.

```
recv(Transmission t){
    //checks the boolean value "dataWaiting" from the packet that it shares
    with the Receiver Thread
    //if true, it fills the given transmission object with the data and sets
    the "dataWaiting" variable to false
    //The Receive thread waits for incoming data. If "dataWaiting" is true
    the thread will wait for it to be false and fills the packet with bytes coming
    from the RF layer,
    //it then sets the boolean "dataWaiting" to true

    //The number of bytes received, or -1 on error.
}
```