SerialForwarder v 1.1

Bret Hull – October 10, 2001

Purpose:

The SerialForwarder application is intended to allow multiple clients to communicate with a mote operating as a base station. Operating on the host PC connected to the mote via a serial port, the SerialForwarder provides a bi-directional packet stream to the clients on the network. This project was conceived with three main goals in mind:

(1) The serial port is a limited resource that can only be opened by at most one application. The SerialForwarder is intended take control of the serial port and act as a proxy between the attached mote and client applications.

(2) The method of communication between a mote and a host PC may change over time. In addition, data collection may be easier from the TinyOS simulator rather than from a live mote network. Rather than have each client application handle the possibly varying communication requirements, the SerialForwarder provides a consistent socket based abstraction for connecting to the motes. Client applications can be simplified and can easily take advantage of changes to the communications architecture.

(3) Client applications need not be running on the PC physically connected to the mote. In many cases, base stations will be distributed over a large area, making physical access rather inconvenient. In addition, client applications may want to pull data from multiple sources, rather than just one mote network. Since the SerialForwarder is socket based, clients can be anywhere in the network and receive multiple streams of data.

Theory of Operation:

The SerialForwarder can be configured to forward data from one of three sources: (1) internally generated dummy data, (2) the TinyOS Simulator, or (3) the serial port. Selection of the dummy data source results in the forwarding of packets, the payload of which is an arbitrary sequence of bits; this source serves as an easy way to test out network connectivity. If the TinyOS simulator is chosen as a source, the SerialForwarder creates a listen thread which waits for the simulator to connect (meaning, the SerialForwarder must be started before the simulator), and then forwards to clients packets generated from the simulated mote network. Finally, if the serial port is chosen as the data source, bytes are read from the serial port (which can be specified on the command line) and are forwarded to clients once a sufficient number of bytes (as defined by the packet size) have been read. Once a source has been specified and after initialization completes, the SerialForwarder creates a server socket to listen for connections from client applications. Once a client connects, it immediately receives a real time stream of packets from the data source. Clients are free to connect and disconnect at will. When no clients are connected, the data source is closed and packets are no longer read until a client reconnects.

Note: the TinyOS Simulator data source is not currently implemented

Getting Started:

The SerialForwarder is a tool written in Java that relies upon the javax/comm library for access to a PC’s serial ports. If the javax/comm libraries are not installed, serial port access will not work.

1. Go to the tools directory in the NEST distribution: `nest/tools`
2. type ‘`make serialforwarder.jar`’
3. type ‘`java -jar serialforwarder.jar`’

The SerialForwarder has several command line options:

-no-gui	no GUI; all output directed to terminal window
-source [serial | dummy | sim]	specifies which of the three possible data sources the SerialForwarder should use (default: serial)
-port [port #]	where port # is the port on which the SerialForwarder should listen for incoming connections (default: 9000)
-comm [serial port]	specifies the serial port to open (default: COM1)
-packetsize [size]	the size of the packets read from the data source (default: 36)
-quiet	do not display any messages
-verbose	display extra messages

Example:

To start the SerialForwarder listening on port 9001, from COM2, with a packet size of 15, and without the gui, type:

```
java -jar serialforwarder.jar -no-gui -port 9001 –source serial -comm COM2 -
packetsize 15
```

To start the SerialForwarder listening on port 9000 (default), reading dummy data, with packet size of 36 (default), and with the gui, type:

```
java –jar serialforwarder.jar –source dummy
```