Wireless Embedded Systems and Networking

Lab Day 3:
Part 1: IP access using standard client programs

Lab Assistant: Jaein Jeong
University of California, Berkeley

Today’s Topics

• Set up IP routing to nodes.

• Ping nodes.

• Using IP level service for nodes.
  – Telnet or TCP, UDP/IP application.
Set up routing to nodes

- By default, sensor nodes have IPv4 address of 10.97.0.0/16.

- Add a route between Ethernet and LoWPAN.
  - Linux: `ip route add 10.97.0.0/16 via <gateway IP>`
  - Windows: `route add 10.97.0.0 mask 255.255.0.0 <gateway IP>`
  - E.g. `route add 10.97.0.0 mask 255.255.0.0 192.168.0.2`

- Exercise 3-1: type the routing command for your environment.

Ping nodes

- Step 1: Find IP address of a node
Ping nodes

• Step 2: Type ‘ping <IP address>’.

$ ping 10.97.85.128

Pinging 10.97.85.128 with 32 bytes of data:
Reply from 10.97.85.128: bytes=32 time=124ms TTL=32
Reply from 10.97.85.128: bytes=32 time=134ms TTL=32
Reply from 10.97.85.128: bytes=32 time=143ms TTL=32
Reply from 10.97.85.128: bytes=32 time=136ms TTL=32
Ping statistics for 10.97.85.128:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
       Minimum = 124ms, Maximum = 143ms, Average = 134ms

• Exercise 3-2: Ping one of the nodes in your network.

Telnet to services on ports

<table>
<thead>
<tr>
<th>Port 7 (echo)</th>
<th>Port 11 (sysstat)</th>
<th>Port 30 (raw sensor reading)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ java EchoClient 10.97.85.128</td>
<td>$ java SysstatClient 10.97.85.128</td>
<td>$ java SensorReadClient 10.97.8.172</td>
</tr>
<tr>
<td>Connected to echo server</td>
<td>Connected to sysstat server</td>
<td>Connected to sensor reading server</td>
</tr>
<tr>
<td>Hello, world.</td>
<td>pltfm : telosb</td>
<td>Hum : 0</td>
</tr>
<tr>
<td></td>
<td>proc : MSP430F1611</td>
<td>Temp : 0</td>
</tr>
<tr>
<td></td>
<td>eui64 : 00173B000C834627</td>
<td>PAR : 0</td>
</tr>
<tr>
<td></td>
<td>version : 2.0.0.11713</td>
<td>builton : 1178639115</td>
</tr>
<tr>
<td></td>
<td>uptime : 6504</td>
<td>built : 1178639115</td>
</tr>
<tr>
<td></td>
<td>cputime : 30</td>
<td>mV : 2999</td>
</tr>
<tr>
<td></td>
<td>radtime : 73</td>
<td>radtime : 73</td>
</tr>
<tr>
<td></td>
<td>mV : 2999</td>
<td>mV : 2999</td>
</tr>
<tr>
<td></td>
<td>wdt : 0</td>
<td>wdt : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port 10 (leds)</th>
<th>Port 15 (netstat)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ java LedsClient 10.97.85.128</td>
<td>Connected to netstat server</td>
</tr>
<tr>
<td>Connected to LEDs server</td>
<td>Lnk TX PTX RX OVH</td>
</tr>
<tr>
<td>109  52  527  115</td>
<td>Net TX RTX DRP OVR</td>
</tr>
<tr>
<td>72   0    0     0</td>
<td>Lnk     TX     PTX     RX     OVH</td>
</tr>
<tr>
<td>Addr  Hops  RSSI</td>
<td>0: 0:0 -40</td>
</tr>
<tr>
<td>0: 0:0 -40</td>
<td>1: 22897 1 -58</td>
</tr>
<tr>
<td>1: 22897 1 -58</td>
<td>2: 10954 1 -52</td>
</tr>
<tr>
<td>2: 10954 1 -52</td>
<td>Rte     Addr  Hops  RSSI</td>
</tr>
</tbody>
</table>
Telnet to services on ports

- Syntax: telnet <node IP address> <port>
- Exercise 3-3: What are the raw sensor readings when you telneted to one of the nodes?

```
$ telnet 10.97.6.172 30
Hum : 0
Temp : 0
PAR : 0
TSR : 2
Vcc : 4019
ADC0 : 1180
ADC1 : 1419
IO0 : 0
IO1 : 0
IO2 : 0
Swtch: 0
```

netcat – what can we do with it??

- netcat can pull information from a host at certain port.
- Syntax: netcat <IP address> <port>
- Exercise 3-4: What are the RSSI reading to the parent node and the first alternate parent node when you netcated to a node?
Custom IP application

- With TCP, UDP/IP Programming
  - Open a TCP or UDP socket using API.
  - Send and receive lines of characters as a message / response.

```java
try {
    Socket theSocket = new Socket(hostname, 30);
    networkIn = new BufferedReader(new InputStreamReader(theSocket.getInputStream()));
    BufferedReader userIn = new BufferedReader(new InputStreamReader(System.in));
    out = new PrintWriter(theSocket.getOutputStream());
    System.out.println("Connected to sensor reading server");
    String reply = networkIn.readLine();
    while (reply != null) {
        System.out.println(reply);
        reply = networkIn.readLine();
    }
} // end try
```

Wrap-up

- Time permitting start on Embedded Programming