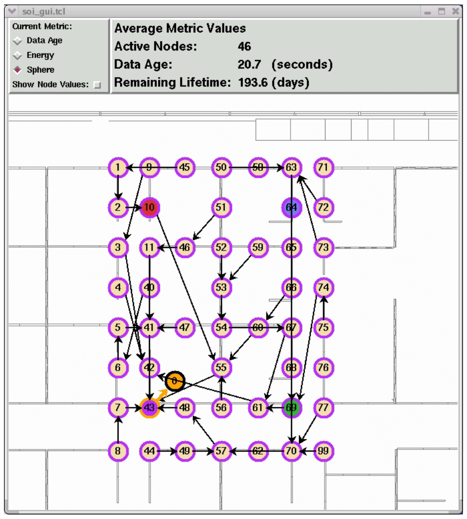
Sensor Bridge

Intel Research and Development

1. Introduction

The Sensor Bridge is software that provides a transparent bridge between sensor networking elements over an IP network. This software is designed to work together with the SensorMesh TinyOS application, which uses the MeshInterfaceM component to transparent multiplex between the radio and UART layers.

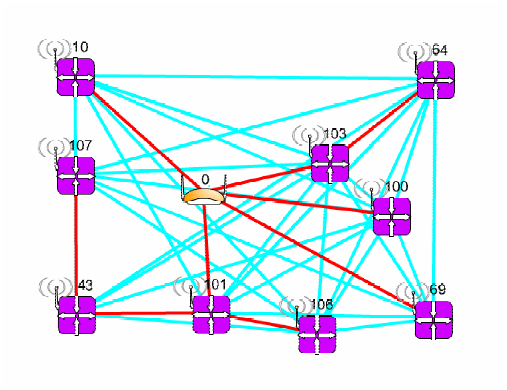
The Sensor Bridge is intended to run on one or more PC-class nodes that are attached to sensor nodes. The PC-class nodes may communicate over any network, including an Ethernet, an Internet, or an ad hoc 802.11 network, allowing the underlying sensor network to be both scalable and physically distributed. For example, consider the sensor network pictured below.



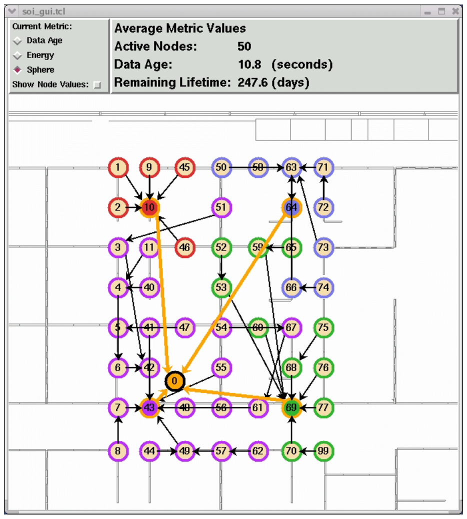
Average Metric Values
Active Nodes: 46
Data Age: 20.7 (seconds)
Remaining Lifetime: 193.6 (days)

Each node is generating data and sending it over a multihop path to the sink node (node 0).

This sensor network could be integrated with an ad hoc 802.11 mesh network, such as the one pictured below.



Note that five of the nodes in the above two networks have the same node ID. If these nodes were connected, then the 802.11 network would create an overlay for the sensor network. Sensor nodes that are also connected to the overlay would then be one hop away from each other. Since each node is trying to send data back to the sink node, and the sink node is on the overlay network, the sensor nodes that are connected to the overlay network will form a star topology, centered at the gateway node. The resulting topology is shown below.



Current Metric:
- Data Age
- Energy
- Sphere

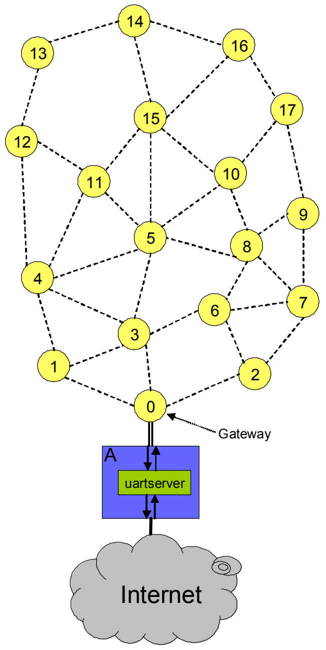
Show Node Values:

Average Metric Values
Active Nodes: 50
Data Age: 10.8 (seconds)
Remaining Lifetime: 247.6 (days)

Note that the sensor network is completely unaware of the presence of the 802.11 network. From the point of view of the sensor network, nodes 10, 43, 64, and 69 are just good ways to get to the sink node. No changes to the routing protocol are required. Similarly, the overlay network is not aware of the sensor network. It just provides a transport for encapsulated sensor network packets.

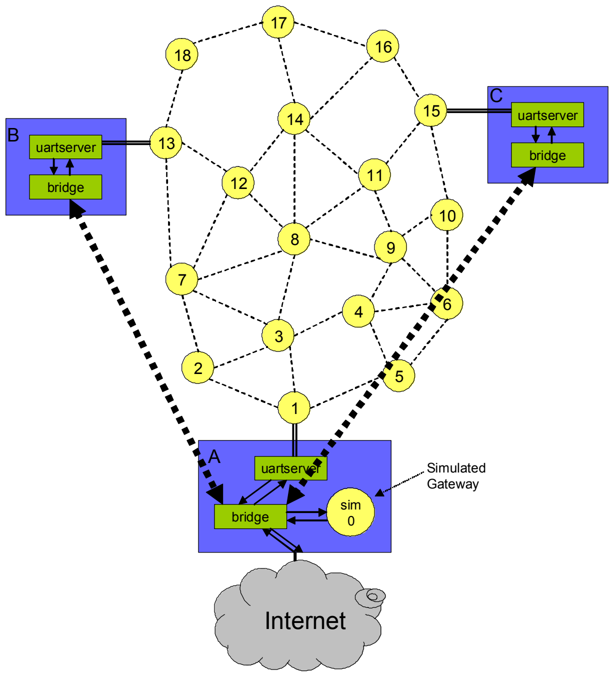
2. Architecture

A typical sensor network consists of a set of sensor nodes, one of which provides a gateway to the IP network, as shown below.



Assuming a DSDV-based network, the gateway will initiate route updates, allowing other nodes in the network to find a path to that node. The gateway node is connected via a serial link to a PC-class node (node A). Using our “uartserver” software, the sensor packets can be served to IP clients on an Internet.

The Sensor Bridge software can be added to this network to allow other points of exit from the sensor network, as shown below.



To maintain the same sensor network topology, with a single sink node, the gateway node is now simulated on the PC-class node (node A). Each of the PC-class nodes (nodes A, B, and C) run the "uartserver" software to read packets from the serial port and a copy of the bridge software.

The Sensor Bridge software forms a star topology. The root of the star is the node that is running the simulated sensor gateway. This Sensor Bridge runs in "server" mode. The Sensor Bridge on the other nodes runs in "client" mode. The "server" Sensor Bridge follows packets to the other software components as follows:

- Packets received from the Internet are forwarded to the simulated gateway. These packets are presumably control packets to be injected into the sensor network. The packets are received by the sensor gateway on it simulated UART interface, allowing them to be received in the manner they would be received by a real sensor gateway, if the Sensor Bridge were not used.
- Packets received from the sensor gateway's UART interface are forwarded to the Internet. These packets are presumably data delivered from the sensor network.
- Packets received from the sensor gateway's radio interface are forwarded to the local serial port (typically via a uartserver) and to "client" Sensor Bridges. No attempt is made to route the packets to the correct destination, as this is handled by the routing software on the sensor nodes.
- Packets received from the "client" Sensor Bridges or from the local serial port (typically via the uartserver) are forwarded to the sensor gateway. The sensor gateway will receive these packets via its simulated radio interface.

These packet forwarding procedures allow nodes connected to the overlay network to believe they are one hop from the sensor gateway. In addition, the same sensor gateway software that is used in the physical network can be used in the simulator.

3. A Sample Execution

The following commands can be used to create an overlay topology consisting of three nodes (A, B, and C) as pictured above.

On node A, start the uartserver:

```
cd tools/packet_tools
./uartserver 9000 COM1
```

On node A, start the simulated gateway

```
cd apps/SensorMesh
./ProgramPC
export DBG=sim
./build/pc/main.exe 1
```

On node A, start the Sensor Bridge

```
cd tools/SensorMeshBridge
./bridge –s –serial SOCK
```

On nodes B and C, start the uartserver

```
cd tools/packet_tools
./uartserver 9000 COM1
```

On node B and C, start the Sensor Bridge

```
.cd tools/SensorMeshBridge
./bridge –c –serial SOCK –beacon NONE <node-A-ip-addr>
```

After each Sensor Bridge is started, it will automatically connect itself to the uartserver on the local machine. The client Sensor Bridges will connect themselves to the server Sensor Bridge. The server Sensor Bridge will connect itself to the simulated gateway.

Once all of the software has started, you will see route updates flowing from the simulated sensor gateway to the bridge clients and data flowing form the bridge clients through the sensor bridge and out to clients on the Internet. Clients may connect to the Sensor Bridge on port 9001, as is typical of the uartserver. The settings manager application can be used to control the network.

4. Command Line Options

While Section 3 listed commands that can be used start a demonstration of the Sensor Bridge, the Sensor Bridge has a variety of command line options, summarized below.

```
Client Mode: ./bridge -c [options]
       or
Server Mode: ./bridge -s [options]


options:
-serial [COM1|COM2|COM3|SOCK] (default COM1)
-gateIP [ip_addr] (server option, default 127.0.0.1)
-beacon [BROAD ip_addr|MULT ip_addr|NONE] (server option, default NONE)
-beacon [BROAD|MULT|NONE ip_addr] (client option, default BROAD)
-quiet (Will not print every packet in stdout)
```

A detailed description of each command-line options follows:

4.1 Client and server mode

The Sensor Bridge must be started in either client or server mode. The “-s” flag specifies server mode and “-c” specifies client mode. Since the Sensor Bridge is intended to be used in a star topology, only one node should run in server mode, while the others run in client mode.

Note that the choice of client or server mode affects the “-beacon” option described below.

4.2 Serial communication

Each Sensor Bridge connects to the sensor network via the serial port (to which a sensor node is attached). Two different methods of communicating with the serial port are provided. If the option "-serial COM <n>" is given, the Sensor Bridge will read raw packets from the serial port. Each packet is expected to be of default length and without any framing.

An external program can also be used for serial communication. If the option "-serial SOCK" is given, then the Sensor Bridge will open a TCP connection to port 9000 on the local machine. Packet can be sent and received on this connection in hex-encoded ASCII, one packet per line. This method is intended for use with the uartserver, which supports packet framing and adjustable packet lengths.

4.3 Specifying the local address

In Section 3, the simulated sensor gateway was deployed on the same machine as the server Sensor Bridge. The "-gateIP <address>" option can be used to specify the address of the machine where the sensor gateway software is running. The default address is the loopback address. This option is only meaningful for the server Sensor Bridge.

4.4 Beaconing

The client Sensor Bridges use TCP to connect to the server. Several mechanisms are used to help client nodes find the server node.

The first mechanism provides no beaconing, and depends on a static configuration. If the server specifies "-beacon NONE", then clients should specify "-beacon NONE <server-ip>". Clients will connect to the specified server address.

A broadcast beaconing mechanism is provided to allow clients within a broadcast domain (i.e., on a single Ethernet) to find the server. The server should specify "-beacon BROAD <server-ip>", where "<server-ip>" is the address that clients should use to contact the server. The server will periodically send a UDP broadcast packet with the server's IP address. If "-beacon BROAD" is specified, clients will listen for the server's broadcast and use the specified address to open a connection to the server.

A multicast mechanism is provided for use when the clients are not within broadcast range of a server. The client specifies "-beacon MULT" and the server specifies "-beacon MULT <server-ip>." These options are very similar to the broadcast mechanism described above, except that a UDP multicast packet is used. Note that this mechanism requires multicast support on your network.

4.5 Miscellaneous

When the "-quiet" option is not given, each packet is printed to the terminal. This option suppresses that behavior.