Demo Abstract: Interactive In-Field Inspection of WSNs

Matthias Ringwald
Institute for Pervasive Computing
ETH Zurich
mringwal@inf.ethz.ch

Mustafa Yücel
Institute for Pervasive Computing
ETH Zurich
yuecelm@student.ethz.ch

Kay Römer
Institute for Pervasive Computing
ETH Zurich
roemer@inf.ethz.ch

1. INTRODUCTION

We demonstrate a tool to inspect nodes of a deployed wireless sensor network (WSN) in the field. While an operational WSN can be remotely monitored via multi-hop routing, a newly deployed network often does not work as expected (e.g., multi-routing is broken). Here the developer needs to inspect single nodes in the field, carrying along necessary tools such as laptop and multi-meter.

An in-field inspection tool on a compact device does not only simplify the process of collecting information about the nodes' state but also enables the actual users of the WSN to perform routine checks or to upload new firmware versions. Our tool running on a standard PDA allows to:

- Re-task nodes by uploading a new firmware image
- Query system and application specific attributes
- Display the topology of the network and execute commands on nodes

We will first provide an overview on the hard- and software of our prototype, then assess how sensor nodes can be accessed and selected for inspection. Finally, we will give a brief description of the provided services and mention related work.

2. SYSTEM OVERVIEW

For this prototypical implementation we wanted to use hard- and software which is known to work and easy to use while trying to be close to actual sensor networks. We chose the deployment-support network (DSN) application running on BTnode rev3 nodes[1] as an exemplary WSN application to incorporate our new inspection services.

The BTnode is similar to the Berkey Mica2 mote but contains an additional Bluetooth radio module and 256 KByte of external SRAM memory. On this hardware, the BTnut system software provides a Bluetooth stack and other services on top of the non-preemptive multithreaded Nut/OS. It includes event signaling, timer, dynamic memory allocation and POSIX-like streaming I/O.

The inspection tool was developed in Java and runs on Familiar Linux 0.8.3 (prerelease) on an HP iPaq H5450.

3. ACCESSING SENSOR NODES

In order to inspect a particular sensor node, we need to be able to communicate with it. Basically, there are three different ways to access a sensor node:

1. Cable: A physical connection allows for direct communicate with a node. Although this is convenient during development and testing with a limited number (< 20) of nodes in the lab, it gets impractical when the nodes are actually put in enclosures and deployed in the environment or their number increases.
2. Normal radio interface: The radio used for the node's wireless communication can also be used to implement inspection services without extra hardware. By using the same radio, the service communication might interfere with other communication and influence the nodes' operation.
3. Extra radio interface: Adding a second radio overcomes the radio interference problem of the previous approach but introduces a scheduling issue of how and when the additional radio is activated.

The chosen DSN application already provides and maintains a reliable Bluetooth scatternet. In order to communicate with a normal PDA, we chose to use the Bluetooth communication also for the inspection services, hence following the second approach.

4. SENSOR NODE SELECTION

Using wireless communication, the user could interact with a potentially large number of nodes in communication range and has to select a single node or a set of nodes to interact with. This selection can be based on an explicit ID (e.g. "what's this node with ID 17 here doing?"), on a certain feature or a predicate ("Let's check nodes with low battery"), or on their physical location.

Independent from the selection criteria, it is beneficial that a selected node provides some feedback to the user, for example, to identify the physical node which has been selected. Common ways to provide feedback are to flash an LED or to emit audible noise.

In our prototype, we used the Bluetooth inquiry mechanism to collect a list of nodes in range. Among those, one or more nodes can be select based on their MAC address which is printed on the node. Upon selection of a node, it's LED flashes as acknowledgment. Using the topology service described later other nodes from the multi-hop network can be selected for inspection, too.

5. SERVICES

We implemented three basic services: re-tasking, attribute query and topology visualization for the inspection tool. On the BTnode, they are available as separate modules. This had been greatly simplified by the used Bluetooth L2CAP protocol and the use of one Nut/OS thread per service. The main application now can instantiate any combination of services to be available via Bluetooth.

5.1 Re-tasking

Every sensor node has its application stored as a firmware image in the microcontroller's flash ROM. On the BTnodes, a wireless firmware upload can be implemented by downloading the new firmware into the external SRAM. Upon reboot, the boot loader then writes the new image into the flash ROM.

We have implemented an L2CAP service that handles the download and the storage of the new firmware image. The PDA allows to select a firmware image and a set of nodes to which the new image is uploaded.

5.2 Attribute Query

The attribute query service provides access to important system and application information of the sensor nodes. Examples for vital system attributes are battery voltage, available heap memory, number of running threads or the node's uptime. Current sensor readings, their polling interval or network statistics are examples for application-specific attributes.

As a mechanism to provide access to such attributes in a generic way, we created an attribute registration module at which the main application can activate system attributes or register their own. An attribute consists of a human-readable name and an access function which has to return the current value as text. We choose to use text for both attribute names and values to provide access to the attributes without detailed application-specific knowledge. The result from the access function could be the current value of a global variable or the result of a more complex calculation.

The attribute registration does not provide any means of communication. In our implementation, an L2CAP attribute-retrieval thread is used to provide the attribute query service via Bluetooth. The PDA retrieves the list of available attributes from the sensor node and presents it to the user who can then request the value of particular attributes.

5.3 Topology Viewer

The visualization of the network topology is a valuable tool for network maintenance and to debug routing problems. But as the network topology depends on the particular WSN application, the topology information cannot be provided without help from the application.

Obviously, the implementation of this service needs to be adapted to the particular network used. The used DSN did already provide a topology scan function so we could display its tree topology as shown in Figure 1.

6. RELATED WORK

The idea of using a PDA for field-inspections was mentioned before in several publications and a PDA has already been used in the TASK project[2]. The TASK field tool provides the ability to ping a single node, issue a command to turn on an LED or a buzzer or to reset the node.

The Nucleus Sensor Network Management System (SNMS) [3] allows the query of system and application state of nesC TinyOS applications. Its integration with nesC allows to remove attribute names from the firmware image by encoding them as numbers but requires to have the correct decoding information available when the query is performed.

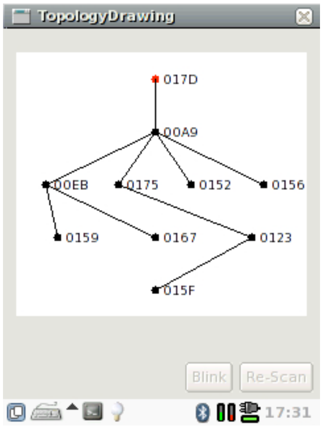

Figure 1: Topology view of the DSN application running on a Linux iPaq H5450.

7. DEMO SETUP

The demo setup consists of a large table with roughly 20 BTnodes and 1-2 HP iPaq H5450 devices.

The provided PDAs allow interested visitors to interact with single nodes or the complete DSN BTnode network. On the first GUI window, the user can perform an inquiry to get a list of available sensor nodes. To perform a firmware download, the set of nodes to be reprogrammed and the new firmware image has to be selected. Then, the firmware is uploaded to the selected nodes one by one.

In the attribute retrieval dialog, the user can select a single node for inspection. After establishing a connection to the node, the PDA displays the list of available attributes. Upon selection of an attribute, its current value is displayed.

The topology mode allows to gather topology information from the DSN network and displays it on the PDA. For this, a connection to one node of the network needs to be established. This view also allows to let all nodes or only single node blink or select a single node for attribute querying.

8. ACKNOWLEDGMENTS

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

9. REFERENCES

[1] J. Beutel, M. Dyer, L. Meier, and L. Thiele. Scalable topology control for deployment-sensor networks. In *Proc. 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05)*, pages 359–363, April 2005.

[2] Phil Buonadonna, David Gay, Joseph M. Hellerstein, Wei Hong, and Samuel Madden. Task: Sensor network in a box. In *EWSN 2005*, pages 133–144, Istanbul, Turkeye, Jan 2005.

[3] Gillman Tolle and David Culler. Design of an application-cooperative management system for wireless sensor networks. In *EWSN 2005*, pages 121–132, Istanbul, Turkeye, Jan 2005.