2008 International Conference on Information Processing in Sensor Networks

Demo Abstract: Sensible Doctor - A Mobile Diagnosis Tool for Wireless Sensor Networks

Seunghun Cha, Hyojeong Shin, Hojung Cha
Department of Computer Science
Yonsei University
Seodaemun-gu, Shinchon-dong 134, Seoul 120-749, Korea
{shcha, hjshin, hjcha}@cs.yonsei.ac.kr

Categories and Subject Descriptors

D.2.5 **[Software Engineering]**: Testing and Debugging—diagnostics, distributed debugging, error handling and recovery, tracing;
C.2.1 **[Computer-Communication Networks]**: Network Architecture and Design—wireless communication

General Terms

Design, Experimentation, Management

Keywords

Diagnosis, Software Fault, Wireless Sensor Networks, Deployment

1. Introduction

Software failures in wireless sensor systems are difficult to debug due to resource constraints and the wireless feature of the deployment. Therefore, it is important to provide developers with relevant information that enables monitoring and analysis of wireless sensor applications before a failure disables a node and makes the node irresponsive.

Sensor networks monitoring applications generally consume resources steadily, so they may influence the performance of sensor networks and increase the traffic load of the networks. Most of the previous work [1,2,3,4,5] on sensor debugging and monitoring has focused on monitoring whole networks. In the real world, however, we often need to monitor a part or subset of the deployed sensor networks.

In this demo, we show a mobile monitoring application, called Sensible Doctor, for sensor network diagnosis. The solution is applied to a restricted region of the network, which surrounds the mobile monitoring device. The network traffic is reduced to cover only the monitoring part of the network by adopting algorithms tailored to mobile monitoring. Sensible Doctor is implemented on the RETOS operating system [6] and provides system information, system profiling, remote application installation and network topology monitoring.

2. System Design

Sensible Doctor extracts information from the sensor network and diagnoses potential problems. The system helps to understand the possible problems the network may suffer from.

2.1 Application Fault Detection

Sensor networks have, in general, no display module to inform users of the status of sensor nodes. Application developers have the normally nontrivial task of debugging their applications. The developer may observe abnormal symptoms from the networks, but the problem can only be detected with partial clues. The observed symptoms reflect various causes of the problems at hand. Any malfunction in the application which has originated from the developer's mistake can be detected by the operating system.

Sensible Doctor supports a safety module which detects illegal operation of the application and reports this information to the kernel. Sensible Doctor manages system information such as the program counter and the stack pointer, so that the system reports the main cause of the fault. This can then be utilized for the debugging process.

2.2 Network Fault Handling

Although a cross-layer approach is common in sensor operating systems, the network stack usually consists of several layers. For any network problem, investigation of the system software components should be implemented to detect if the motes are physically connected, the operating system has enough resources, the wireless links are symmetric, or the routing algorithm is determining a proper routing path, and so on.

Sensible Doctor abstracts the network stacks with four layers: the system layer, link layer, routing layer and application layer. Each layer has a diagnosis tree to determine the cause of networking failure. Sensible Doctor displays the network topology and network traffic with possible causes of packet loss.

978-0-7695-3157-1/08 $25.00 © 2008 IEEE
DOI 10.1109/IPSN.2008.26

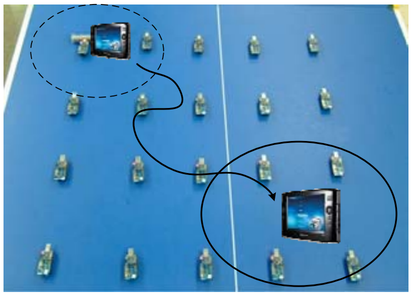
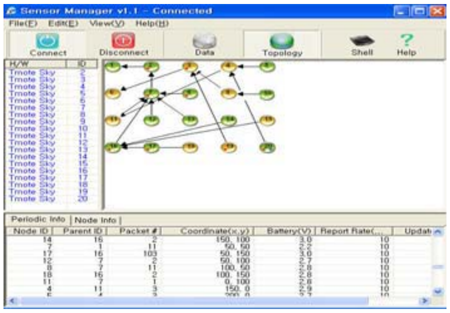


Fig. 1. Demonstration setup


Sensor Manager v1.1 - Connected
File(F) Edit(E) View(V) Help(H)
Connect
Disconnect
Data
Topology
Shell
Help
Periodic Info
Node Info

H/W	ID
Tmote Sky	2
Tmote Sky	3
Tmote Sky	4
Tmote Sky	5
Tmote Sky	6
Tmote Sky	7
Tmote Sky	8
Tmote Sky	9
Tmote Sky	10
Tmote Sky	11
Tmote Sky	12
Tmote Sky	13
Tmote Sky	14
Tmote Sky	15
Tmote Sky	16
Tmote Sky	17
Tmote Sky	18
Tmote Sky	19
Tmote Sky	20

Node ID	Parent ID	Packet #	Coordinate(x,y)	Battery(V)	Report Rate(...	Updat
14	16	2	150, 100	3.0	10	
7	1	11	50, 50	2.2	10	
17	16	103	50, 150	3.0	10	
12	7	2	50, 100	2.7	10	
8	7	11	100, 50	2.8	10	
18	16	2	100, 150	2.8	10	
11	7	1	0, 100	2.8	10	
4	11	3	150, 0	2.9	10	
5	4	3	200, 0	2.7	10	

Fig. 2. Snapshot of the monitoring tool

2.3 System Profiling

Sensible Doctor continuously monitors the internal behavior of the system, such as resource requests of the application, the arrival of external events and sleep scheduling. The information is utilized for sophisticated functionalities, such as power management or real-time scheduling. Sensible Doctor gathers and accumulates the system information, which is provided to the profiling abstraction layer. Any system component is able to reference the information. For example, the diverse power management algorithm archives the previous activity of an application and decides which devices to shut down. Real-time policy also requires system status, such as the response time of sensor devices and the run time of each procedure, to achieve their functionalities. The initial version of Sensible Doctor observes information such as patterns of network communications, requests to sensor devices and the computational costs of applications.

2.4 Mobility Support

Mobile monitoring is useful for most large-scale sensor deployments where the monitoring process of the entire network would cause severe network problems and too much complexity. By wandering though the network, a mobile monitor gathers information only on a particular region of the network. The monitoring node triggers sensor nodes surrounding it to report their data. The reporting group dynamically changes while the monitoring node wanders though the network. Messages for handling the reporting group should therefore be reduced.

Mobile monitoring is an example of a mobile sink application. Movement of the mobile device frequently leads to changes in the reporting path. Sensible Doctor deals with connections between neighbor nodes and also the position of the sink node, which is the mobile monitoring node. The tracing code is reduced, as only the sensor nodes surrounding the mobile monitoring node observe and update movement of the mobile node.

3. Demonstration

We demonstrate the functionality of Sensible Doctor using 1 UMPC and 20 Tmote Sky motes running the RETOS operating system. Figure 1 shows the network topology of the testbed deployment and Figure 2 is a snapshot of the monitoring tool.

Each node periodically delivers system information towards the sink and reports detailed information. The list of applications, resource usage and profile information of each application are retrieved if requested. To demonstrate how accurately and immediately the monitoring tool reflects the mobility of the mobile sink, we alter the topology while the UMPC moves around. Since Sensible Doctor detects application malfunctions, an errant application is installed into the system to demonstrate the functionality.

4. ACKNOWLEDGMENTS

This research was supported by the National Research Laboratory (NRL) program of the Korean Science and Engineering Foundation (No. M10500000059-06J0000-05910) and the MIC (Ministry of Information and Communication)'s ITRC (Information Technology Research Center) program (IITA-2008-C1090-0801-0015).

REFERENCES

[1] Veljko Krunic, Eric Trumpler, Richard Han. *NodeMD : Diagnosing Node-Level Faults in Remote Wireless Sensor Systems*. MOBISYS, 2007

[2] Matthias Ringwald, Kay Römer, Andrea Vitaletti. *Passive Inspection of Sensor Networks*. DCOSS, 2007

[3] Gilman Tolle, David Culler. *Design of an Application-Cooperative Management System for Wireless Sensor Networks*. EWSN, 2005

[4] Yonggang Jerry Zhao, Ramesh Govindan, Deborah Estrin. *Residual Energy Scan for Monitoring Sensor Networks*. WCNC, 2002

[5] I. Jung, H. Cha. *RMTool : Component-Based Network Management System for Wireless Sensor Networks*. CCNC, 2007

[6] H. Cha, S. Choi, I. Jung, H. Kim, H. Shin, J. Yoo, C. Yoon. *RETOS: Resilient, Extensible, and Threaded Operating System for Wireless Sensor Networks*. IPSN, 2007, http://retos.yonsei.ac.kr