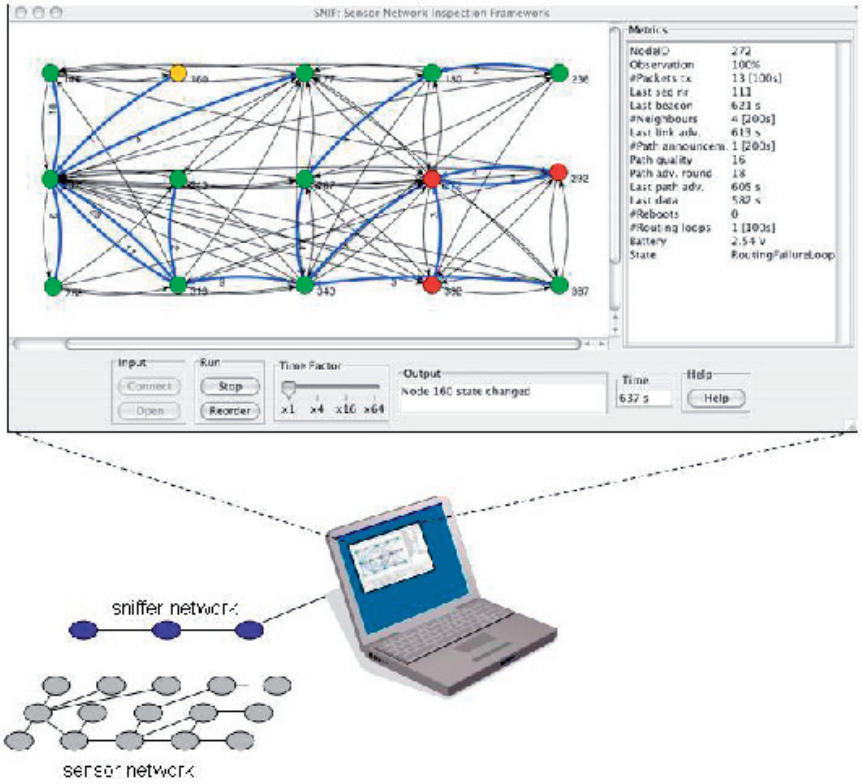
THE DEPLOYMENT OF SENSOR NETWORKS

When deployed in a real-world setting, many sensor networks fail to meet application requirements even though they have been tested in the lab prior to deployment. For example, many sensor networks have been reported to deliver only between 40% and 80% of the sensor data they are expected to produce.



The reasons for such failures can be manifold, including hardware problems (e.g., condensing humidity causing short circuits in a sensor), software bugs (e.g., timing problems that cause the microcontroller to reboot frequently), and networking problems (e.g., communicating nodes fail to wake up concurrently due to excessive clock drift caused by extreme temperature variations). Unfortunately, these problems are often not encountered during predeployment tests, because the environmental conditions that trigger these problems are hard to simulate in the lab.

TO HELP ENGINEERS

The project "Deployment of Sensor Networks", jointly executed by the Computer Engineering and Distributed Systems groups at ETH Zurich, is therefore concerned with the provision of concepts and tools to help engineers develop and deploy sensor networks that meet application requirements. Besides testing aspects and programming models, one of the recent focus areas of the project is inspection: given a deployed sensor network that fails to meet application requirements, inspection is concerned with identifying and locating failure causes in-situ on the deployment site. As we will see below, the Deployment-Support Network [2], also developed within this project, is a key technology for inspection.

Resource limitations of sensor networks make inspection a hard problem. Firstly, the strict need for energy efficiency results in sensor network designs that expose very little information about the system state to an outside observer: every bit of extra information a sensor node exposes about itself increases energy consumption. However,

without sufficient information about the system state it is hard to identify failure causes. Secondly, the scarcity of system resources (e.g., memory, CPU cycles, network bandwidth) makes instrumentation of sensor networks for inspection problematic: adding logging software and monitoring protocols to the sensor network consumes a substantial fraction of the system resources and hence results in a system behavior that may differ substantially from the original, unmodified system.

PASSIVE INSPECTION

Based on this observation, we have recently focused on passive inspection, where "passive" refers to the fact that a sensor network does not have to be modified to allow inspection. In particular, we install a small number of additional nodes alongside the sensor network to overhear the messages exchanged among sensor nodes. These "sniffer" nodes use a second, robust communication channel (e.g., a powerful radio operating on a different frequency or cable) to send the overheard messages to a base station, which generates a globally ordered message trace. As the lifetime of the sniffer network is relatively short, energy and resource constraints are not a major issue here.

By studying typical protocols used in sensor networks, we found that a great deal of information about the state of the sensor network can be inferred from a message trace. For example, sensor nodes typically broadcast so-called beacon messages at regular intervals that contain the sender address and a sequence number. Using these messages, sensor nodes learn about the existence of neighboring nodes and their link qualities. By analyzing beacon messages in a message trace, we can detect, for example, dead sensor nodes (those from which no messages are received) or node reboots (upon reboot, the sequence number is reset). By analyzing routing and data messages in the message trace, we can even learn about routing topologies or detect the existence of network partitions without even touching the sensor network.

NETWORKING PROTOCOLS

However, some important bits of information cannot be inferred this way. For example, in many cases it is impossible to decide if a certain node has successfully received a certain message. An interesting direction for future research is therefore the question how to design networking protocols such that a maximum of information can be gained by analyzing message traces without introducing much protocol overhead.

A key challenge with trace analysis is dealing with incomplete information, as the sniffer network may fail to overhear some messages, so that we need ways to decide whether a message was not sent at all or whether the sniffer network just failed to overhear the message. Fortunately, sequence numbers, timing analysis (for messages sent in regular patterns), and statistical techniques alleviate this problem in practice.

THE SNIF SOLUTION

We are currently building a framework called SNIF (Sensor Network Inspection Framework) [1] to support automated online trace analysis. To deal with the lack of standardized protocols, SNIF offers a packet description language to parse packet contents for a large class of protocols. Furthermore, SNIF offers a data stream abstraction to flexibly process the resulting stream of parsed messages. To inspect a sensor network, a SNIF developer has to define or customize data stream operators to detect certain failure states of sensor nodes. For example, there are pre-defined operators to detect dead nodes and nodes that are partitioned from the

sink. SNIF also offers a graphical user interface to display the current network state. In the screenshot on page 2, node color indicates failure state (green=ok, yellow=warning, red=failure), thin arcs between nodes indicate neighborhood relationships, and thick arcs indicate the inferred routing topology. Our prototype implementation of SNIF uses the BTnode-based Deployment Support Network [2] as a sniffer network. Using this prototype, we could show that SNIF can reliably detect typical failure states even if the sniffer network fails to overhear a large fraction of messages.

Kay Römer, Matthias Ringwald

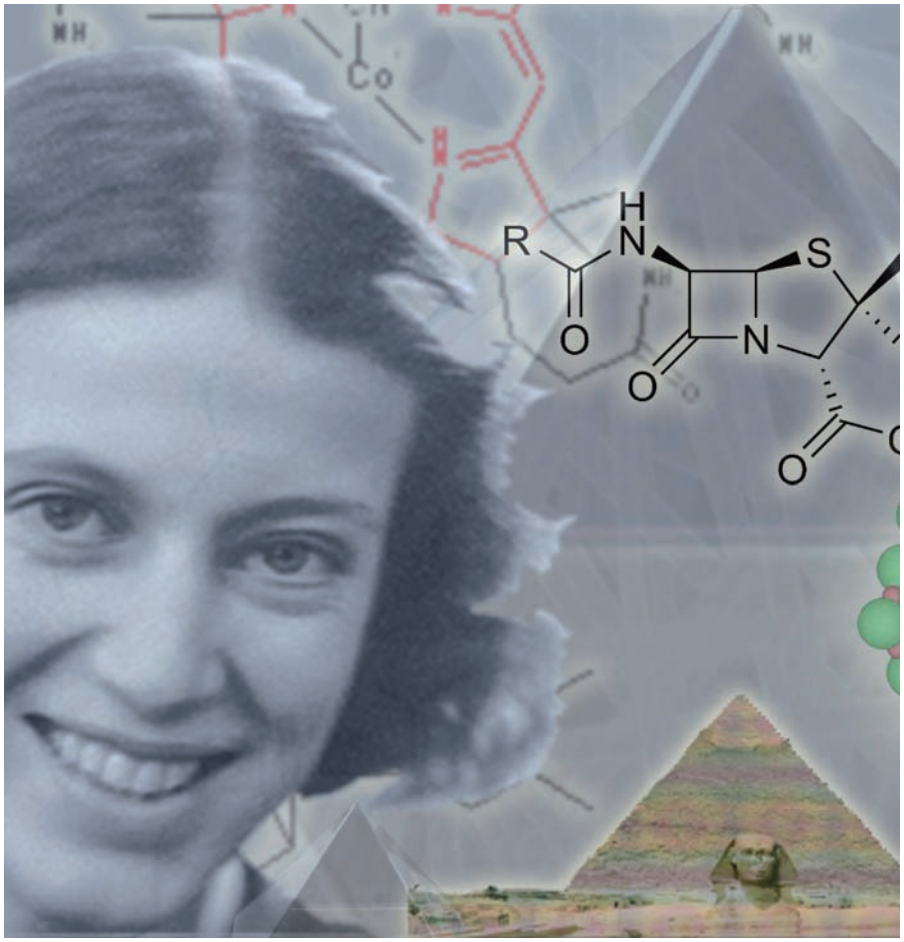
[1] **http://www.vs.inf.ethz.ch/snif/**
[2] **http://www.btnode.ethz.ch/Projects/Jaws/**

National Competence Center in Research
Mobile Information and Communication Systems

MICS

NEWSLETTER

FEBRUARY 2008



A poster of the exhibition on "Women in Science" to be held at EPFL, from March 25th until May 31st