Demo Abstract: Development and Test with the Deployment-Support Network

Jan Beutel, Matthias Dyer, Mustafa Yücel, Lothar Thiele
Computer Engineering and Networks Lab
Swiss Federal Institute of Technology (ETH), Zurich
CH-8092 Zurich, Switzerland
Email: beutel,dyer,yuecel,thiele@tik.ee.ethz.ch

***Abstract*— The Deployment-Support Network (DSN) is a new, minimal invasive methodology for developing and testing wireless sensor networks (WSN) in a realistic environment. A device under test, e.g. a target sensor network is augmented with a redundant wireless backbone network and a second set of nodes to carry out the testing logic. Enabling a developer to test a live application scenario on the actual target hardware on-site in an actual deployment while maintaining correct operation of the system enables the coordinated development as well as reproducible and realistic profiling of the behavior of sensor network nodes.**

I. Introduction

With the advancement of sensor networks beyond the pure proof of concept experience, applications and real-world deployments are getting increased attention. However it is still surprisingly difficult to assemble the components and technologies developed for sensor networks into a functional whole; especially when resources are limited and a minimum performance should be guaranteed [1]. Often, a first iteration of a sensor network system will perform only poorly and a number of subsequent iterations are necessary to achieve a satisfactory level of confidence in the application.

The Deployment-Support Network (DSN) [2], [3] is a tool for the development, debugging and monitoring of distributed wireless embedded systems in a realistic environment. The basic idea is to use a second wireless network consisting of so-called DSN-nodes that are directly attached to the target nodes. The DSN provides a separate reliable wireless backbone network for the transport of debug and control information from and to the target-nodes. However, it is not only a replacement for the cables in wired testbeds but it also implements interactive debugging services such as remote reprogramming, RPC and data/event-logging on the DSN nodes clearly separating debugging and testing logic from the experiment.

II. DSN–Architecture

A. Overview

Figure 1 shows an overview of the different parts in a DSN-system. On the right hand side is the *DSN-node/target-node pair* that is connected via a short cable or interface board, referred to as the *wired target interface*. DSN-nodes are battery-operated wireless nodes with a microcontroller and a radio-module, similar to the target-nodes.

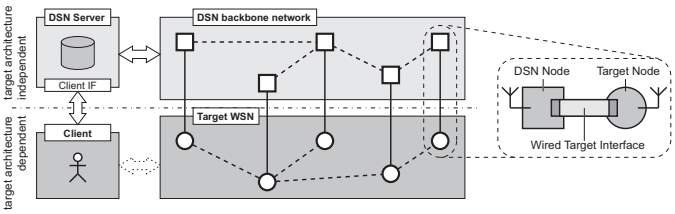

Fig. 1. Conceptual view of a DSN-system with five DSN-node/target-node pairs and a DSN server that can be accessed via the web.

In the center of the figure, there is a conceptual view of the Deployment-Support Network with the two separate wireless networks: the one of the DSN-nodes and the one of the target-nodes. The network of the DSN-nodes is a automatically formed and maintained multi-hop backbone network, that is optimized for connectivity, reliability and robustness.

The *DSN-server* is connected with the DSN-backbone-network and provides the client interface, over which the client can communicate and use the implemented DSN-services. The *client* is a target-specific application or script. The information flow goes from the client over the DSN-server to the DSN-nodes and finally to the target nodes and vice versa. The DSN-server decouples the client from the target WSN both in time and space. In particular, data from the target nodes are stored in a database and can be requested anytime, and commands can be scheduled on the DSN-nodes. Separation in space is given through the client interface that allows for an IP-based remote access.

B. Implications

Using this approach a minimum invasive but yet high degree of observability is guaranteed on a target sensor network either in development or under test. Extending the classical infrastructure (serial port or Ethernet) based WSN testbed [4], [5] with wireless connectivity increases it's versatility and mobility; making arbitrary node placements possible in a real deployment area while still in an early phase of development. Here the MoteLab approach using fixed Ethernet back-channel is clearly inflexible as even in buildings with Ethernet cabling in place, access to this security critical infrastructure is often limited if not impossible. The distribution of the centralized control and logging logic onto multiple DSN-nodes adds the

the scalability and reduces the interference caused by an in-system observer or in-network reprogramming [6]. Many, ultra low-power duty-cycle applications that transfer only a minimum payload or that use nodes with very limited resources can not support in-network reprogramming, which the DSN approach clearly alleviates. Experience has shown that today quite often application developers do not consider to integrate debugging and reprogramming facilities into their applications due to the increased application complexity. Here, the DSN approach with a clear separation of concerns into the target sensor network functions on the one side and the deployment-support functions on the other side (i) allows operation of a target sensor network application with it's correct timing properties, (ii) allows for frequent configuration changes without the burden of a heavy debugging infrastructure and (iii) simplifies the integration effort necessary to integrate all test and debugging infrastructure in a monolithic application.

III. Development and Test: Basic DSN-Operation

The implementation of the Deployment-Support Network is based on the BTnode platform. In order to set up a testbed the BTnodes have to be programmed with the DSN application software and attached to target node devices using an interface board (see Figure 2). These node pairs are then distributed according to the test specification. They can be used either with fixed power using simple USB cabling or be powered from batteries to allow quick changes in the locations and independence from infrastructure (wall outlets). An access node is connected to a PC running the DSN-server software that takes care of all communication to and from the Deployment-Support Network. Optionally and depending on the target sensor network "under test", a server or access node for the target network (e.g. a tmote connect) can also be used to monitor and control the target application.

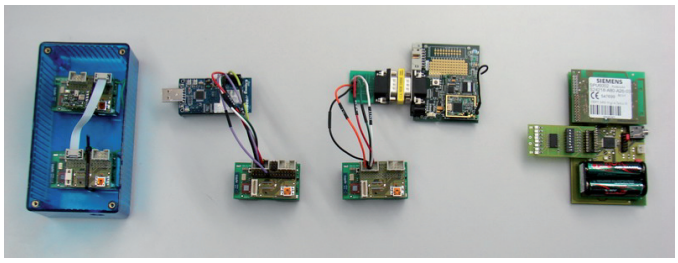

Fig. 2. DSN-node/target-node pairs: BTnode, Tmote Sky, TinyNode, A80.

The DSN-server offers a single point of access to the DSN resource using XML-RPC and logs all interaction to a database. It can be controlled via a web-based interface (see Figure 3). Using this web interface, the DSN nodes are controlled, status is monitored, commands and software updates are sent to the target nodes (via a DSN/target-node pair), and logged data can be retrieved. Using the interface to the DSN-server, a developer can attach tools that suit his current needs in the development, deployment and testing process. For example, software for the assessment of the RF channel and radio performance can be uploaded to different target devices and operation is logged for a given period of time. After completion of a test sequence the logged data is retrieved and analyzed. If desirable the process can be repeated, using modified target software, or by controlling the operation of individual target devices, e.g. to inject coordinated faults into a system under test.

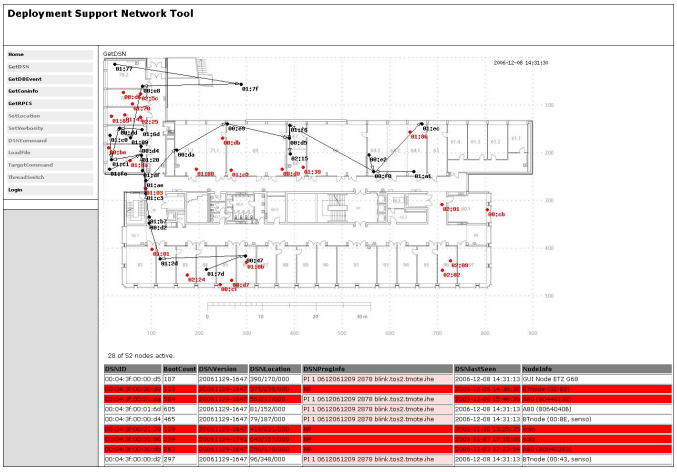

Fig. 3. Web-based control interface for the Deployment-Support Network.

In other scenarios, fine grained logging down to interrupt granularity might be of concern, e.g. when debugging low level driver software, or long term logging and monitoring for application validation purposes.

In different case-studies [7] the Deployment-Support Network has proven to be a versatile and powerful tool that can be used in a number of different ways in the development and deployment process of wireless sensor networks.

References

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A macroscope in the redwoods," in *Proc. 3rd ACM Conf. Embedded Networked Sensor Systems (SenSys 2005)*, pp. 51–63, ACM Press, New York, 2005.

[2] J. Beutel, M. Dyer, M. Hinz, L. Meier, and M. Ringwald, "Next-generation prototyping of sensor networks," in *Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004)*, pp. 291–292, ACM Press, New York, Nov. 2004.

[3] 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)*, pp. 359–363, IEEE, Piscataway, NJ, Apr. 2005.

[4] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A wireless sensor network testbed," in *Proc. 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05)*, pp. 483–488, IEEE, Piscataway, NJ, Apr. 2005.

[5] L. Girod, J. Elson, A. Cerpa, T. Stathapopoulos, N. Ramananthan, and D. Estrin, "EmStar: A software environment for developing and deploying wireless sensor networks," in *Proc. USENIX 2004 Annual Tech. Conf.*, pp. 283–296, June 2004.

[6] J. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004)*, pp. 81–94, ACM Press, New York, Nov. 2004.

[7] M. Dyer, J. Beutel, L. Thiele, T. Kalt, P. Oehen, K. Martin, and P. B., "Deployment support network - a toolkit for the development of wsns," in *Proc. 4th European Workshop on Sensor Networks (EWSN 2007)*, pp. 195–211, Springer, Berlin, Jan. 2007.