Poster Abstract: Next-Generation Prototyping of Sensor Networks

Jan Beutel, Matthias Dyer, Martin Hinz, Lennart Meier, Matthias Ringwald*
Swiss Federal Institute of Technology (ETH) Zurich
8092 Zurich, Switzerland
beutel@tik.ee.ethz.ch

ABSTRACT

Large-scale deployment of sensor networks is more and more becoming an issue to researchers and industry alike. The recently revised BTnode architecture provides two wireless radios and facilitates the interconnection of heterogeneous devices. Apart from offering interesting new opportunities in using multi-frontend devices in sensor-network research, this architecture is optimally suited for deployment-support networks as introduced in the following.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Management, Design, Experimentation, Standardization

1. INTRODUCTION

With the increasing development of large-scale Wireless Sensor Network (WSN) applications, the coordinated development and deployment of sensor network devices are becoming an issue of increasing importance [2, 7]. Independent researchers have reported that when moving away from the engineer's desktop and beyond numbers of 10–20 nodes, deployment and testing become increasingly hard, and simulation will not solve all problems. While algorithms, system models, device architectures, and programming abstractions have been investigated for quite some time now, not much has been achieved in the area of deployment support or even a concerted design and deployment flow that allows for stepwise refinement and reliable monitoring of systems. Coordinated methods and tools for WSN deployment are missing today.

With our approach presented here, we push the limit for large-scale prototyping from virtualization [3] to real-world deployment. We first introduce the novel concept of the BTnode rev3 platform, and then describe the operation of our deployment-support network.

*The work presented here was supported 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.

SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
ACM 1-58113-879-2/04/0011.

2. MULTI-FRONTEND BTNODES

The BTnode prototyping platform [1, 2] has been recently revised, mainly (i) to make use of a new Bluetooth subsystem and (ii) to incorporate a second, low-power radio, namely the Chipcon CC1000. This radio is also used on the Berkeley Motes, so that the new BTnode is a twin, both of the Mote and the old BTnode. Both of its radios can be operated simultaneously or be independently powered off completely when not in use, considerably reducing the idle power consumption of the device (see Fig. 1).

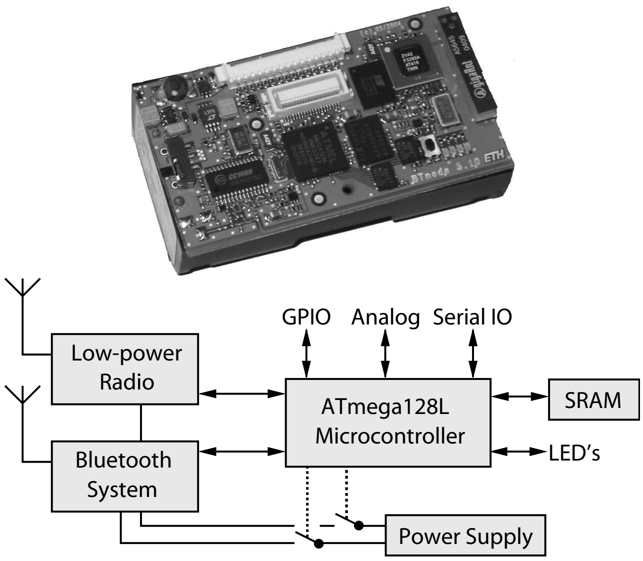

Figure 1: The BTnode rev3 system overview.

This new architecture opens new opportunities to interface between ultra-low-power devices like the Berkeley Motes and larger devices such as Bluetooth-enabled appliances [4], or to investigate duty-cycled multi-frontend devices with wake-up radios [6] or bandwidth–power–latency trade-offs.

The BTnode hardware can run TinyOS [5], or the BTnode system software that is based on an Ethernut kernel and allows cooperative multithreaded applications with standard C programming.

3. DEPLOYMENT-SUPPORT NETWORKS

Classic approaches to develop and deploy WSNs use serial cables for program download, control and monitoring. Although successful in lab setups, this approach is limited due

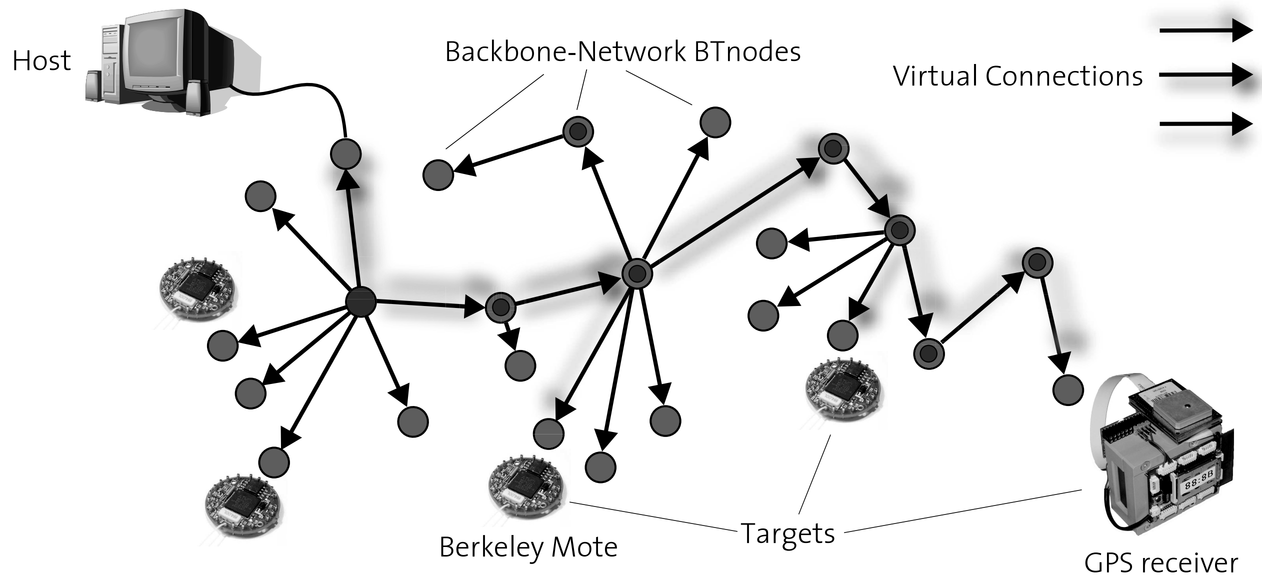

Figure 2: The BTnode deployment support network.

to scalability issues and completely infeasible for deployment in the field.

Our approach is a wireless serial-cable replacement offering reliable and transparent connections by attaching a BTnode to every target, e.g. a Berkeley Mote, in a certain deployment scenario. Using the target WSN itself for control and monitoring is not an option since any additional traffic could disturb the actual WSN application, and in the early phases of development the target WSN is too unreliable.

The BTnode devices construct and maintain a multihop backbone network. A host, e.g. a PC, can then attach to any of them and open a virtual connection to an arbitrary BTnode in the network. The host can then communicate both with this node and with its attached target through the virtual connection. So far, we have implemented remote programming, target control and monitoring (see Fig. 2); other BTnode–target operations are also possible.

In this application, a multihop network formed by Bluetooth Scatternets is most suitable due to the reliable link layer, multiplexing, QoS capabilities and rather high bandwidth of Bluetooth. This bandwidth is necessary when tunneling the aggregate traffic of multiple virtual connections. A simple, robust and distributed algorithm constructs a tree topology. The tree topology is self-healing, i.e. it automatically takes care of joining and leaving nodes.

The virtual connection replaces the direct host–target serial cable. The packet switching at every BTnode is based on ATM virtual circuits and automatically forwards traffic to the appropriate connection. Thus, no routing is necessary on our tree topology.

Benefits of this deployment-support network are scalable and transparent connections to arbitrary target devices. It is plug-and-play, self-configuring and requires no alteration of the target system. In demonstration setups, we have successfully built tree topologies spanning 30–40 BTnodes and having multiple virtual connections with data-rates up to 57.6 kbit/sec.

Opportunities for future work are redundant connections to manage link and node failures more quickly, data aggregation, optimal topology control and routing.

4. REFERENCES

[1] J. Beutel, O. Kasten, and M. Ringwald. BTnodes - a distributed platform for sensor nodes. In *Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys 2003)*, pages 292–293. ACM Press, New York, November 2003.

[2] J. Beutel et al. Prototyping wireless sensor network applications with BTnodes. In *Proc. 1st European Workshop on Sensor Networks (EWSN 2004)*, volume 2920 of *Lecture Notes in Computer Science*, pages 323–338. Springer, Berlin, January 2004.

[3] L. Girod et al. EmStar: A software environment for developing and deploying wireless sensor networks. In *Proc. USENIX 2004 Annual Tech. Conf.*, pages 283–296, June 2004.

[4] J.L. Hill et al. The platforms enabling wireless sensor networks. *Communications of the ACM*, 47(6):41–46, June 2004.

[5] M. Leopold, M.B. Dydensborg, and P. Bonnet. Bluetooth and sensor networks: A reality check. In *Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys 2003)*, pages 103–113. ACM Press, New York, November 2003.

[6] E. Shih, P. Bahl, and M. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *Proc. 6th ACM/IEEE Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2001)*, pages 160–171. ACM Press, New York, September 2002.

[7] R. Szewczyk et al. Lessons from a sensor network expedition. In *Proc. 1st European Workshop on Sensor Networks (EWSN 2004)*, volume 2920 of *Lecture Notes in Computer Science*, pages 307–322. Springer, Berlin, January 2004.