Networked Embedded Systems Lab
Prof. Marco Zimmerling

Development of a Network Simulator to Analyze Communication Strategies in WSN

Master Project Proposal

Wireless sensor networks (WSN) consist of resource-scarce devices with little computational power and a strong focus on energy efficiency. Their flexibility enables new applications; however, wireless communication uses a shared medium subject to interference and is less reliable than communication over wires. Many wireless communication protocols coordinate their packet transmissions to avoid collisions (interference) between multiple transmitters, which can often lead to packet loss. In contrast, more recent protocols based on synchronous transmissions (ST) allow transmissions to overlap within a specific time window. It has been shown that ST-based communication protocols outperform traditional protocols in terms of latency, reliability, and energy efficiency while greatly simplifying the coordination in the network. The timing requirement forces ST-based protocols to operate in successive time slots, where a node decides to either transmit a packet or listen for an incoming packet in each slot. These decisions are, among other things, the key for efficient communication with ST.

The goal of this thesis is to develop a network simulator for ST-based communication protocols in Python and implement and analyze different communication strategies. A simulation environment has the advantage that different strategies can be implemented and analyzed quickly, without all the lower-level hardware, software, and timing challenges that one has to cope with on actual embedded hardware. The simulator should support different (possibly random) network topologies and simulate the behavior of the various nodes for each slot. To test different communication strategies, the behavior of the nodes should be easily configurable such that each node can potentially use a different strategy. As a baseline, the simple communication pattern of Glossy [1], an ST-based communication protocol, should be implemented within the simulator and compared to other strategies, which can also be learning-based. The execution traces should be recorded, and evaluation scripts need to be created to analyze and compare the different strategies according to typical metrics, for example, communication latency and reliability.

Requirements

- Good coding skills in Python.
- Interest in embedded systems and networking.

Contact

- Fabian Mager, fabian.mager@tu-dresden.de
- Prof. Marco Zimmerling, zimmerling@cs.uni-freiburg.de

References

[1] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 2011.