

Re-Schedule

A Test Scheduling Framework for Re-Mote

This document describes a master thesis project to design, implement, and evaluate a test scheduling framework for deeply embedded networked systems. The framework will provide a way to configure tests in terms of topology, mote platforms, and job control and automating tasks related to postmortem analysis and tracking the state of the system under test. The test scheduling framework will build on the infrastructure of the Re-Mote Testbed Framework by extending it to support job execution.

Project Tasks

Job creation and configuration

Since the intended users of the system is both experienced researchers as well as students taking their first course on sensor networks, the system should allow powerful configuration while remaining easy to use. For very complex jobs, dynamic configuration should be supported by providing tools for topology management and job control.

Job control

To facilitate more advanced experiments with timed events such as simulated mote failure or injection of data and packages an interface for controlling jobs is required. Several possibilities will be investigated from enabling users to include a calendar of predefined events to allowing the users to use a job control language.

Testbed resource management

The scheduler for the framework will involve a resource manager with the aim of configuring the resource utilization of the testbed. Currently, Re-Mote allows interactive usage and does not support reservation. The design should therefore contain an analysis of what mote properties should be included in mote selection and how often used network topologies can be chosen.

Protected job execution

While users should be able to run jobs locally, it should be possible to also submit jobs for execution on the testbed server infrastructure. Because jobs can contain control code and other parameters and many jobs may run simultaneously, the system should separate job execution by running them in a sandbox and actively monitor jobs and intervene if their resource usage is deemed harmful.

Project Specific Goals for Learning

1. Give a comprehensive analysis of the requirements and technical problems of testing (*applications developed for*) deeply embedded networked systems.
2. Based on theory and similar projects, evaluate methods for building a robust test scheduling framework.
3. Design and implement a test scheduling framework.
4. Evaluate different methods for dynamically configure tests with the aim of extending the set of tested features.