Motivation & overview

Goal: to ease sensor network programming in the large

- Design:
  - **Macroprogramming** the network with **Abstract Task Graph (AtAG)**
  - Tool-assisted **network** model specification
  - Automated **mapping** of tasks to network nodes

- Deployment:
  - Multiple firmware **configurations**
  - Over-the-air **task setup**

- Maintenance:
  - **Express reliability requirements** as part of the model
  - **Runtime assurance**: check the satisfaction of these requirements
  - Automatically **reallocate** tasks on failure

Features

- **Network** layout and sensor node property description
- **Task** definition and wiring
  - A task is an arbitrary node-local chunk of application functionality with a fixed interface
  - Nonfunctional **requirement** specification
    - Constraints on min PDR, max delay, etc. (pictured)
  - Integration with simulator
  - Run-time support through middleware

Architecture

- **Main components:**
  - Web interface
  - Task allocator daemon
  - Gateway daemon
  - Interface with a network simulator
  - Sensor net middleware library

- **Distributed microservice architecture**
- **Communication**: JSON over HTTP

Technical highlights

- **Builds on existing tools**
  - WSN software (e.g. Contiki and Cooja)
  - Gecode: for constraint solving in the task mapping algorithm
  - Node-RED: for UI look-and-feel

- Experimental **Glossy-based scheduler** for the control plane
- **Contiki extended** with:
  - Reliable mesh protocol
  - Glossy / ContikiMAC timesharing
  - JSON-over-HTTP interface to Cooja

- **Probabilistic variables** (e.g. link qualities) represented by probability distributions
- Automatically sets up extra tasks on redundant network nodes in case reliability requirements without them cannot be satisfied

Evaluation

- **Task allocation performance** with instances from [Pathak 2010]
  - (objective function: minimize average energy consumption)

- **Task setup performance** in 2x2, 4x4, 6x6, and 8x8 node grid networks