WSN-Erlang: a Functional, High Level Approach to WSN Development

Applications executed in distributed environments, composed of heterogeneous devices with different capabilities and limited resources, interconnected using short and unreliable wireless links.

WSN challenges

Micro-programming (state of the art)
Programmer has to decompose the problem and develop code for each node. This approach does not hide the complexity of WSNs.

Macro-programming (state of the art)
Developer can see the whole network as a single entity. This approach reduces the complexity of development, but solutions can be applied on a limited number of scenarios.

Problem
Code and fix - No reusability - Difficult to maintain

Solution
WSN - Erlang (concurrency, heterogeneity, bit streams, fault-tolerance)

Current Erlang characteristics:
- complex runtime system: hiding network protocols and configuration
- large number of libraries: from basic functionalities to distributed databases and CORBA compatibility
- high hardware requirements

For WSN scenarios we need:
- low hardware requirements
- minimum number of libraries
- simpler runtime, for example the network stack

Current work
(1) further reducing the hardware requirements of our interpreter, and
(2) exploring how to adapt testing and model checking approaches developed for Erlang to WSN-Erlang.