ABSTRACT

Licentiate thesis

A Language-Based Approach to Protocol Stack Implementation in Embedded Systems

Author: Yan Wang, IDE, Halmstad university

Embedded network software has become increasingly interesting for both research and business as more and more networked embedded systems emerge. Well-known infrastructure protocol stacks are reimplemented on new emerging embedded hardware and software architectures. Also, newly designed or revised protocols are implemented in response to new application requirements. However, implementing protocol stacks for embedded systems remains a time-consuming and error-prone task due to the complexity and performance-critical nature of network software. It is even more so when targeting resource constrained embedded systems: implementations have to minimize energy consumption, memory usage etc., while programming efficiency is needed to improve on time-to-market, scalability, maintainability and product evolution. Therefore, it is worth researching on how to make protocol stack implementations for embedded systems both easier and more likely to be correct within the resource limits.

In the work we present in this thesis, we take a language-based approach and aim to facilitate the implementation of protocol stacks while realizing performance demands and keeping energy consumption and memory usage within the constraints imposed by embedded systems. Language technology in the form of a type system, a runtime system and compiler transformations can then be used to generate efficient implementations. We define a domain specific embedded language (DSEL), Implementation of Protocol Stacks (IPS), for declaratively describing overlaid protocol stacks. In IPS, a high-level packet specification is dually compiled into an internal data representation for protocol logic implementation, and packet processing methods which are then integrated into the dataflow framework of a protocol overlay specification. IPS then generates highly portable C code for various architectures from this source. We present the compilation framework for generating packet processing and protocol logic code, and a preliminary evaluation of our compiled code.