Comparative Modelling of Underwater Wireless Sensor Systems

Dr. Xianhui Che, Gordon Dickers
Faculty of Applied Design and Engineering
Swansea Metropolitan University, UK
{ xianhui.che; gordon.dickers}@smu.ac.uk

Abstract

The challenges for designing underwater wireless sensor systems include power consumption, reliability of communication links, network routing protocol, distributed localization and time synchronization. Common traffic patterns for sensor networks include the collection of data from all sensing devices and transmission of data to the base station through the wireless network. Most sensor networks consist of a protocol stack from physical layer, routing layer, to the transport layer and application layer. Although there are many protocols existing on each layer, not all of them are feasible for deployment in sensor networks as the requirements for different systems can vary.

On top of the network aspects of the wireless sensor system, modeling and simulation of the underwater nodes have to take into account environmental factors, in that elements such as temperature, water clarity, turbidity, etc., can easily affect the quality of the signal transmission. A broad variety of different simulation tools can be used to simulate key characteristics of wireless sensor networks. Each of these tools has its specific advantages and disadvantages and often the selection of the tool is mainly based on the experience of the researcher rather than on rational arguments. An overview of the different tools and simulation environments with their particular advantages and disadvantages will be established in the paper. Comparisons will be made by using an identical system scenario that is specially designed for coastal monitoring purposes. The system to be modeled forms a static small scale network infrastructure.

The designated wireless sensor will use radio frequency technology as the underwater physical transmission medium, as radio signal offers more advantages than acoustic and optical signals in the shallow water conditions where there is a high level of sediment and aeration in the water colum. The chosen routing protocol for this system is built in an operating system called Contiki which is used to control individual motes which make up the nodes of the wireless sensor network. The Contiki operating system has several features which are attractive to sensor network developers including a small memory footprint. It is open source based and includes a light weight protocol stack designed for low power networking. A simulation tool, called Cooja, has also been made available by the Contki developers to facilitate the development and testing of sensor networks and it is this simulator, amongst others, which will be evaluated in the paper.

Cooja is a cross level simulation tool written in Java. It can simulate sensor networks simultaneously at different levels, including the operating system level and the network (application) level. At the operating system level Cooja simulates the target system by executing the native Contiki based sensor code (including network protocols and user applications) whilst at the network level it allows the creation and modification of radio models which makes possible communication between sensors. Cooja also allows developers to extend its functionality by adding custom plugin components which might, for example, log data or monitor network events.

Another modeling tool will be used to compare the simulation's performance with Cooja. OPNET Modeler is a commercial network simulation software package which provides a platform for modeling and simulating network applications. It allows users to design and study communication networks, devices, protocols, and applications with good flexibility and scalability. OPNET Modeler is a discrete event simulator and is widely used by the network R&D community. OPNET Modeler's object-oriented modeling approach and graphical editors mirror the structure of actual networks and network components, providing an intuitive user interface.

Cooja and OPNET are compared regarding the ease of implementing the scenarios, resource consumption, collecting delay metrics, throughput as well as comparability of the results. The required protocol is built as part of the Contiki operating system but the radio channel and related system I/O needs to be defined and implemented within Cooja, while with OPNET the simulated physical layer radio channel and protocol have be to constructed and implemented from scratch. Based on the measured simulation efficiency and other identified criteria, suggestions will be made as to whether it is advantageous to use a modeling tool that has the built-in protocol, or to use a different tool to model the entire system.