AquaNet: A Network Solution to Underwater Sensor Networks

Zheng Peng[1], Zhong Zhou[1], Jun-Hong Cui[1], Zhijie Shi[1], Lee Freitag[2]
[1] Computer Science and Engineering Department, University of Connecticut, Storrs, CT 06269
[2] Woods Hole Oceanographic Institution, 266 Woods Hole Road, Woods Hole, MA 02543

1. INTRODUCTION

Underwater sensor network (UWSN) is a wireless communication system that consists of battery-powered vessels, sensors nodes and a variety of devices. The fundamental differences between UWSN and terrestrial networks are the channel and signal characteristics. Unlike ground networks, high frequency electromagnetic wave doesn't work well in the water due to the conducting nature of the medium. Acoustic communication becomes the most commonly used technique and brings new challenges [2] to this area because of high attenuation and long propagation delay.

Recent years witness an increasing research interests and progresses in this area. Both academic researchers and engineers made a lot of efforts and many applications, network protocols and devices have been introduced. Most of them are strongly application driven and are not compatible with each other. To make the communication system more successful, there is need to provide a general architecture for underwater sensor network. Ref. [1] is a good starting point that designed an underwater architecture loosely based Open System Interconnection (OSI) model [5].

An underwater architecture should consider the characteristics of the network and make it extendable and flexible enough to accommodate the rapid development of applications and underlying hardware. We propose Aqua-Net, a network framework that follows a layered structure while permitting significant optimization. Different from [1], it supports cross-layer design by introducing a translucent vertical layer that is accessible for all applications and protocols. It's layered structure makes easy integration among implementations of different researchers. And we define an abstract layer, or narrow waist [3], that allow device and protocol developments to proceed apace. Aqua-Net is a valuable platform that will facilitate the process of application development.

2. NETWORK ARCHITECTURE

Our goal is to design an architecture that is extendable, user friendly and upgradable. This means that new modules should be able to be integrated into the platform without too much trouble. It inexplicitly requires an open architecture. The system should also be friendly for debugging. Debugging is always a headache for system software. An architecture is a system that may become more complicated during its evolvement. An error in the software may get notoriously difficult to detect. It's very critical to make the system easy to debug. Furthermore, Aqua-Net should be upgradable. As time goes by and feedbacks collected from users, protocol stack may need to have modifications to better satisfy the needs of applications.

2.1 Design philosophy

2.1.1 *Cross-Layer Design*

Any design reflects special properties of the target system. Most networks, such as the Internet, follow an architectural structure. However, a strictly-layered design is not flexible enough to deal with the dynamics of underwater environments and may prevent performance optimizations. Network protocols have to be very efficient to survive underwater environment. This makes cross-layer design a popular methodology among researchers. Note that cross-layer design should not be a non-layered or single-layered design, which compromises the overall ability to provide continuous growth in scale and diversity of the whole system.

Cross-layer design is important to sensor networks for many reasons. Firstly, system energy consumption is affected by all layers. On the other hand, all layers should co-operate to reduce overall system energy consumption. Cross-layer design is a way to make such cooperation possible. Moreover, network protocols need a bigger picture of the system. This requires some parameters be accessible to multiple layers. However, according to [8], in networks with high attenuation channels, all cross-layer design, can only improve throughput by at most a constant factor. And layered design could be order-optimal.

The Aqua-Net adopts a cross-layer architecture which offers an alternative to the pure layered approach that promotes stricter local interaction among protocols in underwater nodes. Useful cross-layer feedbacks will be utilized by other layers to improve system performance.

2.1.2 *Lowering of the Waist*

Architectural design provides abstract layers for different level users. This makes it easier and possible to implement a system that could be used for a variety of applications and networking devices. There should be a set of shared interfaces and components to bridge the gap between applications and hardware. The most essential component is referred as "narrow waist", just like Internet Protocol (IP) for Internet. This idea is first introduced by [3]. Depending on the properties of the network, the location of "narrow waist" varies. Ref. [3] and [7] proposed a Sensor-net Protocol (SP) to be the "narrow waist" of sensor network. We further argue that this waist could be even lower to Media Access Control (MAC) layer in underwater sensor networks.

Internet is a "network of networks" and designed to provide end-to-end communication among heterogenous autonomous systems. IP naturally becomes the bridge, or narrow waist, of Internet. Similarly, for underwater network, there also exists such narrow waist. However, underwater networks are more application driven and have more security concerns. It doesn't even need to interopertate among different network systems. The Ad-hoc feature also implicitly focuses more on single-hop best-effort communication. MAC layer thus emerges as a good candidate of the narrow waist of underwater sensor networks.

2.2 System Architecture

The system contains both hardware and software. For hardware, We use Micro-Modem [9] as the communication device and Gumstix [4] as the controller. In term of of system software, Embedded Linux [6] is adopted as the operating system and our protocol stack is running on top of it.

Figure2 shows the structure of the protocol stack. Apparently we are not using system socket. Instead, we developed a pseudo BSD socket interface. So end user is still able to use similar interfaces for their implementation. So one of the key problems is, how to emulate BSD socket for ordinary users. Another problem is how to provide the service to the user, or in other words, how to establish communications between emulated BSD socket to our protocol stack. We will see how these are handled in the following section.

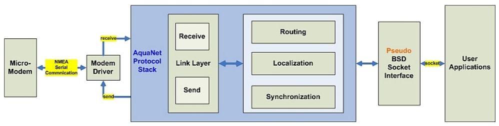

Figure 1: System Architecture

3. IMPLEMENTATION

3.1 Hardware platform: Gumstix

Underwater sensor network is a challenging area. Latest technologies, including software and hardware, should be adopted to improve overall system performance. Advanced algorithms are proposed to reduce energy consumption but require more computation power from processors and support from advanced operating systems. Gumstix [4] is the choice as a result of continuous search for energy efficient, compact, high performance and inexpensive devices.

3.2 Operating System: Embedded Linux

Embedded Linux is the operating system designed or optimized for devices that have limited resources, such as battery life, computational power and storage capacity. It's an open source platform that could be customized by users to suit the need of a specific application. As a variant of Linux, embedded Linux provides similar, if not the same, features and has abundant resources. Researchers don't need to invent every wheel they want, instead, there are a lot of libraries and tools available to use right out of the box. In all, embedded Linux is an advanced, customizable, developer friendly and ever-developing operating system that is ideal for developing underwater network protocol stack.

3.3 Protocol Stack

The protocol stack is a framework that provides various network services. In other words, both network application and protocol developers rely on the services provided by a protocol stack. There are three interfaces for different purposes.

The interface for network application developers is a pseudo-BSD Socket Interface. The application developers will concentrate on their application logic and don't need to know too much about the underlying network. By providing function calls with similar names as the ones in BSD Socket, developers may be able to compile and run their existing codes with no or minor modifications. Behind the sockets, communication among sockets and other system components are done with Unix Interprocess Communication (IPC) techniques. For network protocol designers, they should be able to pick up their interested packets, do all the necessary processings and usually (but not always) send this packet back to an incoming/outgoing packet queue. This implies the existence of a queue management mechanism. The last interface is the lowest layer that is connected to a real network. In our case, this refers to a serial port and related programs that control it.

Despite all the interfaces, the framework has some routines to coordinate all the components. An important routine is to keep all system information. This is actually a vertical layer in our software architecture which makes system information available to all layers. So cross-layer design is possible in our system.

The implementation of protocol stack includes the following techniques: shared memory, semaphore and multi-threading. Figure2 demonstrates our design of protocol stack.

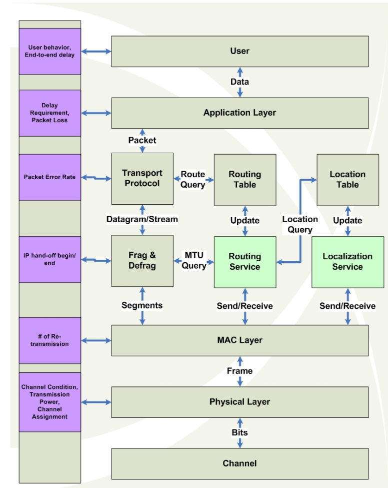

Figure 2: Protocol Stack

4. CONCLUSIONS

In this paper, we proposed and developed Aqua-Net, a network that is architecture scalably and efficiently in underwater sensor networks. It provides a general software and hardware platform for network researchers. A BSD socket-like interface makes it easy for user to implement applications. User developed protocol can also be plugged in to the system. By lowering the "narrow waist", our system can efficiently support a variety of applications and hardware. The main innovation is that while keeping the system flexible and making researchers easy to use or incorporate their own protocols, the system also makes cross-layer design possible by providing system parameters across all network layers. This facilities the process of doing research and field tests.

Several protocols have been implemented within this network architecture. As an on-going work, we are evaluating the performance of the whole system in a coastal environment. A modified ALOHA and two routing protocol will been tested.

5. REFERENCES

[1] M. Chitre, L. Freitag, E. Sozer, S. Shahabudeen, M. Stojanovic, and J. Potter. An architecture for underwater networks. In *Proc. IEEE OceansâĂŹ06 Asia Pacific Conference*, June 2006.

[2] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE Network, Special Issue on Wireless Sensor Networking*, 20(3):12–18, 2006.

[3] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao. Towards a sensor network architecture: lowering the waistline. In *Proceedings of the 10th conference on Hot Topics in Operating Systems*, volume 10, pages 24–24, 2005.

[4] Gumstix Inc. Gumstix - Dream, Design, Deliver, http://www.gumstix.com.

[5] ISO/IEC. ISO 7498: Open systems interconnection - basic reference model, 1984.

[6] Open Source Community. Embedded Linux wiki - A wiki for developers using Embedded Linux, http://elinux.org/.

[7] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica. A unifying link abstraction for wireless sensor networks. In *Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys05)*, 2005.

[8] S. Toumpis and A. J. Goldsmith. Performance, optimization, and cross-layer design of media access protocols for wireless ad hoc networks. In *IEEE International Conference on Communications*, 2003.

[9] WHOI. Micro-Modem, http://acomms.whoi.edu/micromodem/.