Designing a Three-Node Underwater Acoustic Relay Network

Krenar Jusufi, Roy Behymer, Mary Hoyer, and Shengli Zhou

Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, CT, 06269

***Abstract*— Building upon the point-to-point data transmission, we have designed and experimented a three-node underwater relay network. We considered two scenarios for generating messages. In the first scenario, a message input from the graphic user interface can be transmitted to any specified destination in the network. In the second scenario, a motion sensor is attached to one node for continuous motion monitoring. Once an event is detected, an alert message is generated and broadcast to the whole network. We have tested the three-node network in a water tank and in a lake.**

***Index Terms*— Underwater acoustic communication, multicarrier modulation, relay, sensor network**

I. Introduction

Recently, there has been a growing interest in monitoring aqueous environments (including oceans, rivers, lakes, ponds, and reservoirs, etc.) for scientific exploration, commercial exploitation, and protection from attacks. The ideal vehicle for this type of extensive monitoring is a networked underwater wireless sensor distributed system, referred to as the Underwater Wireless Sensor Network (UWSN) [1], [2].

A senior design team at the University of Connecticut has established a computer-based prototype of a multicarrier acoustic modem, and demonstrated the point-to-point data communication both in air and in a water tank [3]. Building upon the point-to-point links, the aim of this project is to establish a three-node relay network, where a message from a source can be transported to any specified destination within the network. We considered two scenarios for generating messages. In the first scenario, a user can type a message with the graphic user interface and specify the desired destination. In the second scenario, a motion sensor is attached to one node for continuous motion monitoring. Once an event is detected, an alert message is generated and broadcast to the whole network.

For the three-node network, we use a static routing table to list the routing path for each possible source-destination pair. When receiving a message, each node checks the source and destination field, and then looks up the routing table to decide the next-hop transmission. We first tested the three-node design in a water tank at the UWSN lab, and then in the Mirror lake at the Storrs campus of the University of Connecticut.

*More details can be found at the website of our senior design project: http://www.engr.uconn.edu/ece/SeniorDesign/projects/ecesd88/

II. Project Demonstration

We demonstrated our project at the ECE senior design day, Dec. 7, 2007, at the University of Connecticut. The testbed is depicted in Fig. 1. Visitors were invited to type messages to be forwarded to any specified destination. Placing an obstacle in the front of the motion sensor would trigger an alert message, which would be broadcast to the whole network. The demonstration was done in the water tank. However, we have also tested the three-node network in the Mirror lake at the Storrs campus, where three nodes were placed on a straight line and with distance one meter apart between nodes. Successful transmissions were achieved in both settings.

III. Design Considerations

A. Packet Design

For convenience, we use byte as the basic unit in designing the packet structure, as shown in Table I. Each packet corresponds to one OFDM block at the physical layer, which supports a data stream of 3.1 kbps with a 5.5 kHz bandwidth [3]. The packet header has 6 bytes. The first three bytes are used for the routing protocol as they specify the next hop address, the source address, and the destination address. The next three bytes are used for data transport, in that they specify the sequence number, how many packets transmitted in one data burst, and how many useful bytes in the last packet. Each packet has 77 bytes as the payload data. The last byte is used for cyclic-redundancy-check (CRC).

TABLE I

THE PACKET STRUCTURE

Field	# of Byte(s):	Purpose:
msgNext	1	Next hop address
msgSrc	1	Source address
msgDest	1	Destination address
msgSeqNo	1	Sequence number
msgNoPack	1	Number of Packets
msgUseful	1	Useful no. of bytes in last packet
msgData	77	Data Bytes in this packet
msgErr	1	End of packet byte

B. Graphic User Interface

The graphical user interface (GUI) is used to have users easily communicate and test the network, as shown in Fig. 2. The interface allows for quick adjusting of several parameters, including:

- Number of OFDM subcarriers

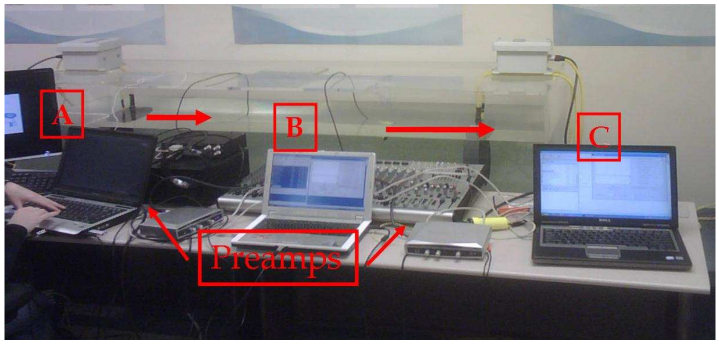

Fig. 1. The three-node testbed using the water tank

- Center frequency (Hz)
- Guard time between data packets
- Synchronization sequence duration
- Pause time between synchronization sequence and first data packet
- Number of packets per transmission
- Correlation threshold
- Channel length
- Trigger value
- Partition buffer

A user can type the message through the GUI. Further, the user can specify the source and the destination for each message. After collecting the relevant inputs, the node looks up the routing table to decide the next hop. Then, it forms a data burst, which may consists of multiple packets, to transmit.

C. Routing

Since there are only three stationary nodes in the network, we use a static routing table that lists the routing paths for all possible source-destination pairs. For example, we place the three nodes on a straight line, and label them as A, B, and C, respectively. If node A generates a message with the source field as A and the destination field as C, then the next-hop is node B in the routing table for the (A,C) source-destination pair. The static routing table needs to be updated when the network topology changes.

D. Motion Sensing

To emulate a surveillance scenario, we attach an ultrasonic motion detector with a USB interface to one node; the motion sensor is shown in the left part of Fig. 3. The motion sensor accurately measures distances as near as 15 cm and as far as 6 m. The motion sensor is set to measure the distance of an object in its path and communicates the data in real time to the computer, taking one data sample per second. We were unable to acquire the motion sensor data directly from the sensor into the MATLAB environment, so we designed a LabVIEW virtual instrument to accept the raw data from the sensor and print it to a text file which is read by the MATLAB code. The LabVIEW layout is shown in the right part of Fig. 3. The MATLAB program compares the object distance from the sensor to a pre-set threshold value. If the object distance from the sensor is less than the threshold, a motion alert message is generated and broadcast from the sensing node to other nodes in the network. A sample alert message would read as follows, “Motion Alert! Time: 11:30:09, Dist: 0.23m”.

IV. Lessons Learned

We next present some useful experience accumulated during the project development.

A. Relay Testing

The main problem we faced during the tests of the relay network was the hardware inconsistency. We used different manufacturers for the three laptops and thus some parameters, such as power level and synchronization correlation threshold, needed to be changed for different laptops. This is due to the inconsistencies on the sound cards from different computer manufacturers. Eventually, we learned which signal levels were best for each laptop as a receiver and as a transmitter. A simple fix would be to purchase laptops from the same manufacturers to avoid this concern.

B. Tank vs. Lake

Tank is a confined area and lake is open. For the testings in the water tank, because of the relatively short distances, there was the concern of self interference due to direct arrivals and relayed messages. During the test, we made sure the second and third messages received were neglected as they may have intervened with our true initial message. For lake testings, the unexpected distress we faced was external disturbances, specifically the flock of ducks that fluttered nearby causing annoyance and interruption to the audio signal.

C. Motion Sensor

Our motion sensor does not interact directly with MATLAB. We could have conducted a better search in finding a motion sensor that would work directly with MATLAB. The introduction of the LabVIEW workaround creates a less compact software package and introduces unavoidable time delays and sluggishness of the overall program.

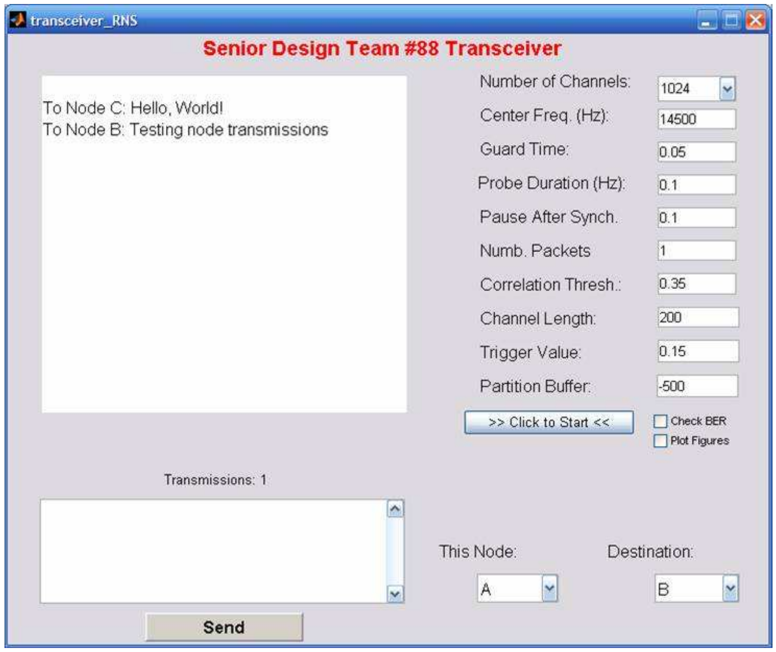

Fig. 2. The graphical user interface

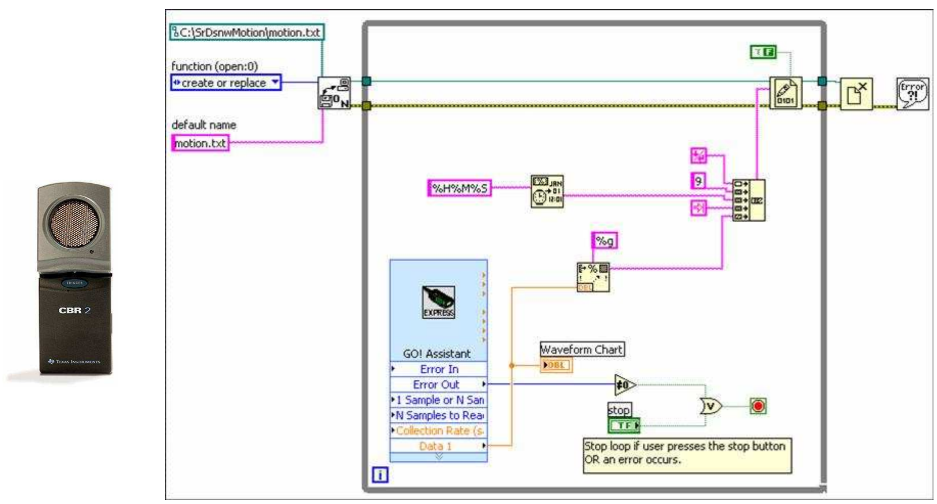

Fig. 3. Left: The motion sensor; Right: The LabVIEW layout for recording the motion sensor output into a text file

V. Conclusions

Our senior design project has implemented a three-node network where one node can transmit messages to any specified destination in the network with possible message relays. In addition, a motion sensor is attached to one node for continuous motion sensing. Once an event is detected, an alert message is generated and broadcast in the network.

Acknowledgements

We would like to thank Mr. Sean Mason for his help in developing the algorithms and testbed, and Mr. Jason Thomas for delivering the presentation at the NCUR conference. We are grateful to the UConn ECE department for the project support, and the ONR YIP grant N00014-07-1-0805 for the travel support.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," *ACM SIGBED Review*, vol. 1, no. 1, July 2004.

[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*, vol. 20, no. 3, pp. 12–18, May-June 2006.

[3] S. Mason, R. Anstett, N. Anicette, and S. Zhou, "A broadband underwater acoustic modem implementation using coherent OFDM," in *Proc. of National Conference for Undergraduate Research (NCUR)*, San Rafael, California, April 2007.