Ethernet Telemetry for Towed Arrays

S. Anantha Narayanan, S. Subash Subramoniam, S. Sham Kumar and V.P Jagathy Raj[*]

Naval Physical & Oceanographic Laboratory, Kochi-682 021, India
[*]School of Management Studies, Cochin University of Science and Technology, Kochi-682 022, India

ananth_prathibha@yahoo.co.in, ssmoniam@yahoo.com, shamkumar04@yahoo.com, jagathy@cusat.ac.in

Abstract: The towed array electronics is essentially a multichannel real time data acquisition system. The major challenges involved in it are the simultaneous acquisition of data from multiple channels, telemetry of the data over tow cable (several kilometres in some systems) and synchronization with the onboard receiver for accurate reconstruction. A serial protocol is best suited to transmit the data to onboard electronics since number of wires inside the tow cable is limited. The best transmission medium for data over large distances is the optical fibre. In this a two step approach towards the realization of a reliable telemetry scheme for the sensor data using standard protocols is described. The two schemes are discussed in this paper. The first scheme is for conversion of parallel, time-multiplexed multi-sensor data to Ethernet. Existing towed arrays can be upgraded to ethernet using this scheme. Here the last lap of the transmission is by Ethernet over Fibre. For the next generation of towed arrays it is required to digitize and convert the data to ethernet close to the sensor. This is the second scheme. At the heart of this design is the Analog-to-Ethernet node. In addition to a more reliable interface, this helps in easier fault detection and firmware updates in the field for the towed arrays. The design challenges and considerations for incorporating a network of embedded devices within the array are highlighted.

Keywords: Ethernet, Towed Array, Embedded network

1. Introduction

Ethernet is the most popular multi-point protocol for data transfer. Today ethernet is omnipresent and it wins hand down as the preferred protocol for an embedded network. During the initial phase of towed array development custom protocols were used for data telemetry. The conversion of the sensor data (not only acoustic but heading, roll, depth, temperature etc.) to ethernet (or any other standard for that matter) within the towed sensor array was a challenging task. Not only the electronics design but also packaging (pressure tolerance) and space were major constraints. But with the miniaturization of electronics, availability of nano connectors and introduction of MEMS (Micro- electro

978-1-4244-9118-6/09/$26.00 ©2009 IEEE

Mechanical System) sensors, it is possible to conceive of arrays of 10 mm diameter.

The transition from a custom protocol based sensor array to a networked one is done in two steps. The first step is to network enable the existing towed sensor arrays. In these systems the data from all the sensors is collected and digitized at a single point. At this stage the Ethernet PCB is introduced which sends the digitized data over ethernet as shown in Figure 1.

The hydrophone signals are weak. Hence the nearer the digitization is to the sensor the better is the performance of the sonar. This is the motivation for the second scheme i.e. conversion of the sensor data to ethernet close to the sensor. This leads to a modular architecture with each module (the Node) lying close to the sensor.

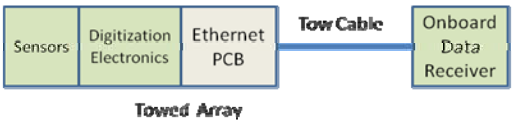

Fig. 1 Network enabling the towed Array

The two schemes are described in the sections below.

2. Ethernet in the Last Lap

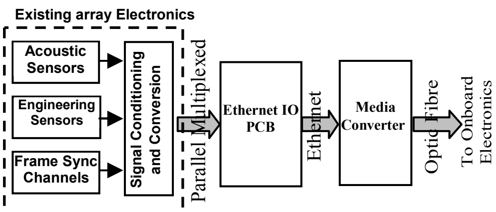

Fig. 2 Towed Array Electronics

The conversion from parallel data to Ethernet can be done in many ways. The easiest way is using microprocessors or microcontrollers to control the conversion process. However due to the speed and space restrictions it is difficult to realize inside the towed array. The faster method is to implement the complete ethernet conversion in hardware. Programmable Logic Devices (PLD) can be used for this purpose. This scheme is easily reconfigurable, fast and easier to design and amenable to rework which is very important during the prototype development.

The design under discussion as shown in Figure 2, uses the Altera EPM1270 Programmable Logic Device and SMSC LAN91C111 Ethernet controller chip. The PLD is the only intelligent device in the PCB.

The Ethernet controller chip is a mixed signal device that implements the MAC (Media Access Control) and PHY (Physical layer) portion of the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) protocol at 10 and 100 Mbps [1]. The total internal buffer size is 8 kilobytes which is the total chip storage for transmit and receive operations. The analog PHY block consists of a 4B5B/Manchester encoder/decoder, scrambler, descrambler, transmitter with wave shaping and output driver, twisted pair receiver with on chip equalizer and baseline wander correction, clock and data recovery and auto-negotiation.

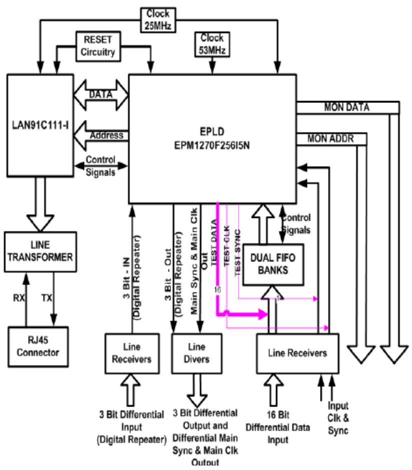

Fig. 3 Block Diagram of Ethernet I/O

The ethernet PCB as shown in Figure 3, converts synchronous parallel data to a packetized serial form. The input to the card is a 16 bit parallel data with clock and sync pulse as shown in Figure 4.

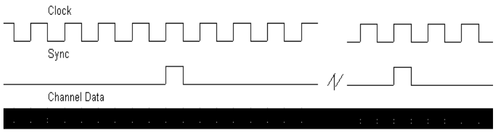

Fig. 4 Input signal waveforms

The input is continuous without any flow control. So input double buffered FIFO (First Input First Out) bank as shown in Figure 5, is used so as to enable the simultaneous read and write. While data is being written into the first FIFO page the data will be read from the second and vice versa.

The data in the FIFO is transmitted as UDP (User Datagram Protocol) over IP (Internet Protocol) over Ethernet packet [2] at 30 Mbps – at this speed you can download an entire movie in just 3 minutes!!!

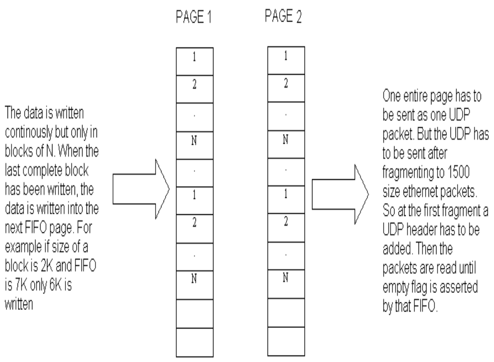

Fig. 5 Double Buffering of Input Data

All these actions are controlled by the PLD logic which is the heart of the PCB. The functions performed by the PLD are

1. FIFO selection, generation of read-write controls and switching
2. Initialization the Ethernet Controller (LAN91C111)
3. Write instructions to load the FIFO data to internal buffer of Ethernet controller
4. Enable Ethernet data transmission and Service Ethernet interrupt

The PLD logic was implemented using VHDL. A simplified block for state machine implemented in the PLD is given below in Figure 6. The actual state machine has close to 400 states to implement this logic. The PLD also acts as glue which is used to route between several devices in the PCB.

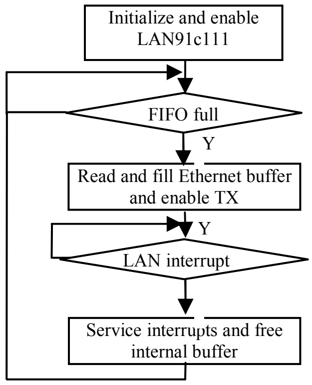

Fig. 6 Simplified State machine Logic

The PLD has to generate the address, data and control signals to write to a particular register in the Ethernet controller over a bidirectional data bus. The amount of VHDL (VHSIC Hardware Description Language) code required for the initialization and packetizing the data is huge with the Ethernet controller initialization alone accounting for 60% of the PLD resources. The communication with the Ethernet controller is implemented as a state machine with branching done depending on the response from the ethernet controller chip [3].

The output of the card is UDP over IP over Ethernet packet [4] as shown in Figure 7. UDP is the transport layer protocol and IP the network layer. It is an unacknowledged connectionless protocol. It is used to identify between multiple applications running on the same machine which are both using the same Ethernet device. IP is a network layer protocol for segmentation and reassembly of multi-fragment packets [5]. The UDP packet size is 8 Kbyte. The higher the input data rate the faster is the rate at which UDP packets are sent. These UDP packets can be directly captured by standard software like LabVIEW UDP read VI (virtual instrument), Wireshark, and Ethereal etc.

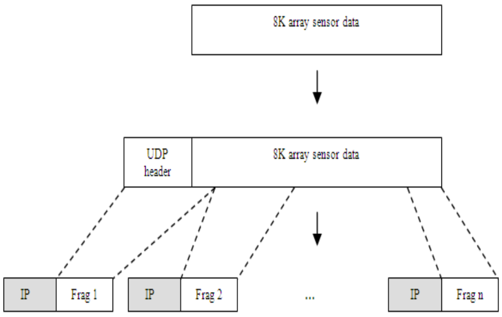

Fig. 7 UDP fragments

The 8K UDP payload is split into several fragments each of which can be transferred over a single Ethernet packet. So there will be 6 fragments since the Ethernet payload is 1500 bytes. The PCB used in this scheme is shown below as shown in Figure 8.

Fig. 8 Ethernet PCB

The ethernet PCB has reduced the onboard receiver to a standard Fibre to ethernet converter [6]. A PC, server or COTS board having an ethernet port can be used as the receiver. Thus the complexity of the onboard electronics also is drastically reduced.

3. Network Based Towed Array

The advantage of using a modular structure to any system is the ease of reconfiguration. There are also more pertinent advantages during testing, integration and production. The variety of hardware comes down and so does the time to realization. Hence the need for realizing the Analog to Ethernet Node based towed array architecture. The scheme under realization is discussed below.

The ethernet node takes the raw data from the transducer; filters, amplifies and digitizes the data and converts it to ethernet. The ethernet packets from all the nodes are sent over the same serial bus as shown in Figure 9. Ethernet data transmission is not hard real time, therefore only the data acquisition is synchronized, not the packet transmission. Sampling synchronization is provided using a common clock signal for all the nodes. Each node stores the data samples and transmits them as a larger block. An alternate method is to use time synchronization protocols over ethernet but this requires additional hardware in each sensor node as well as a time server onboard the platform.

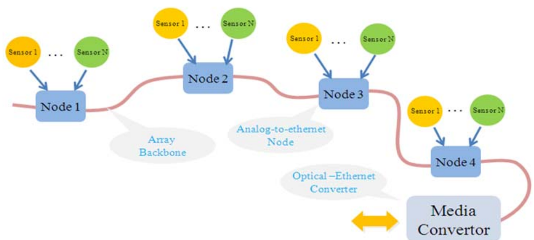

Fig. 9 Network Based Towed Array

The signal conditioning electronics in the node consists of amplifiers and anti-aliasing filters. There are separate ADCs for each channel for the conversion of analog signals to digital samples. Then a network enabled microcontroller converts the sampled data to ethernet. The tricky part here is simultaneous sampling [7]. It will have to be implemented by using hardware like a PLD.

Each node in the network has its own MAC and IP address. The MAC address is fixed for the hardware. The IP address will be determined by the software by its position in the network (Switch settings for each node in serial order). The data from each node will be in UDP over IP over Ethernet format. There will be a time stamp so as to allow the receiver to put together the data from the same instant from all the sensors.

The onboard receiver will be more complicated than the one in the previous scheme. In the present scheme the receiver will have to deal with data coming from multiple sources. From these multiple sources it will have to seek out the data sampled at the same time.

In addition to data from the sensors, the controls to the array from the operator console uses the same interface. This will in turn further reduce the complexity of the system. The modular structure allows for Node health status check by sending a PING like packet to the node. It also gives options like turning off a faulty node, adjusting the gain for individual hydrophones etc. It will also be possible to update the firmware in the Node in the field by using the same interface.

The move towards standard protocols will enable the use of MIL grade COTS (Commercial off the Shelf) components within the system. This will help in reducing the cost and time to realization of the system. Some of the COTS components that can be used are shown below as in Figure 10.

The adoption of this architecture will lead to a drastic reduction in the amount of wiring in the towed system. Our experience shows that the lesser the wires density in the array the better its reliability and performance.

Fig. 10 COTS components

4. Conclusions

The Ethernet PCB has network enabled the existing towed array systems. It has replaced the custom interface with a standard one. This has led to the simplification of the onboard receiver. As a result the reliability and ease of testing and maintenance has increased.

The Network Based Towed Array will be a leap forward in terms of performance and reliability of towed arrays. The major challenges are in the pressure tolerant packaging and miniaturization of the electronics. The modular architecture is a powerful concept with a network of simple ethernet devices dividing the complex task of simultaneous multi sensor real time data acquisition into smaller manageable tasks.

References

[1]. Johnson, Howard W., "*Fast Ethernet: Dawn of a New Network"*, New York: Prentice Hall PTR, 1996

[2]. Tanenbaum, Andrew S., "*Computer Networks",3rd Edition* Prentice Hall of India, 2001

[3]. Catsoulis, John. "*Designing Embedded Hardware"*,O'Reilly,2002

[4]. Black, Uyless. , "*Computer Networks: Protocols, Standards and Interfaces"*, Prentice Hall 1987.

[5]. Stallings, William. "*Networking Standards: A guide to OSI,ISDN,LAN and MAN Standards"*, Addison Wesley 1994

[6]. Gibson, Jerry D. *The communications handbook*, CRC Press 1997

[7]. Bidgoli, Hossein. "*Handbook of Computer Networks Vol.2 : LANs, MANs, WANs, the internet and Global Cellular and Wireless Networks"*, John Wiley and Sons Inc.,2007