Design and Development of a Giga-bit Ethernet Based High Speed Broadband Data Acquisition System for an Underwater Imaging Array

Subash K, Amogh R and Venugopalan P

National University of Singapore, 12A Kent Ridge Road, Singapore 119222

***Abstract*-Ambient Noise Imaging (ANI) has been an active area of research at the Acoustic Research Laboratory (ARL). The ARL under Tropical Marine Science Institute (TMSI) of National University of Singapore (NUS) had designed an ambient noise imaging array in 2003 and tested them in the field. To cater for a high data-rate of 1.6 Gbps, Fibre Channel Arbitrated Loop technology was employed. Although this arrangement satisfied the data-rate requirement, the resulting complexity, size, and huge power requirements of the system along with a lack of reliability still needed to be addressed. The recent availability of relatively low cost, small form-factor, multi-channel off the shelf data acquisition cards and low power PC104+ embedded platforms along with Giga-bit Ethernet technology has been used to address the limitations of earlier system. The use of embedded Linux operating system has helped in improving the reliable operation of the system. In this paper we give a detailed account of the new hardware & software design and implementation followed a performance assessment of the system through evaluation in the field.**

I. INTRODUCTION

A 2D Broadband Ambient Noise Imaging (ANI) array and its associated receiver system had been built and tested in 2003 by ARL and the results have been reported in [1]. Even though the system was able to perform its tasks, there were number of issues that limited a longer and reliable operation of the system in the field. This ANI system was engineered with 54 embedded computers running on windows NTe for data acquisition and over hundred optical transceivers modules for Fiber Channel networking. As a result, the system consumed about 1700 Watts and after accounting for the losses a 2KW power supply was required to power up the system. The high power consumption also resulted in build-up of heat inside the casing, which is held under semi-vacuum. As there were no means for taking the heat outside off the system either through conduction or convection, the electronics had to be switched off for more than an hour after every couple of acquisition schedules spanning a few minutes. The system has also experienced boot-up problems due to corruption of OS which were stored in compact flash disks mounted on each embedded processor. The system was also limited in dynamic range performance as the front-end design did not have the facility to program the signal gain. As the underwater deployment and data collection exercise is highly resource intensive and expensive, it was essential to ensure reliable performance of the imaging system over a longer period of time in the field. A flexible design which would allow the user to change the front-end gain in the field was required to ensure that we get the best dynamic range when operated under different environments. This necessitated redesign of the hardware & software pertaining to the data acquisition and streaming electronics of the imaging array. Recent technological advancements such as energy efficient computing, high-speed networking and availability of high density memory have made it possible to realize a low power, highly reliable imaging array along with a compact receiver system.

As depicted in Fig. 1 the data acquisition system for underwater imaging studies consisted of two main sub-systems, a 2 dimensional sensor array with all the electronics (wet-end electronics) to collect and stream the data and a receiver (dry-end system) at the surface to receive, store and process the data. The wet-end system consisted of 508 broadband acoustic sensors spanning the plane of a 1.4 m diameter disk. The system operated in the frequency band of 25 KHz to 85 KHz.

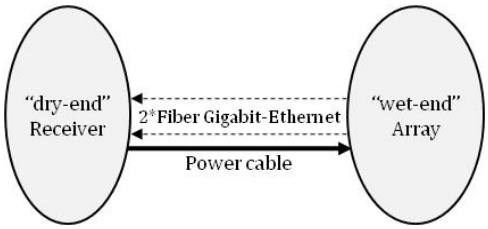

Figure 1. Underwater Data Acquisition system components

These 508 acoustics sensor signals were sampled simultaneously and transferred to the receiver system at the surface. Simultaneous sampling of 508 channels added challenges to the design, but it was an essential requirement to preserve phase information during beamforming. The wet-end and dry-end systems were connected by a 160 meter hybrid cable; electrical wires to feed power to the array electronics and optical cable for data transfer as well as for communications with the array. The following sections describe the design and implementation of the hardware and software architecture of this novel data acquisition system along with a discussion on its performance in the field.

II. HARDWARE ARCHITECTURE

The Hardware of the ANI system consisted of a 512 channel Data Acquisition System. Each sensor signal was passed through a dedicated signal conditioning stage. At this stage a fixed front-end gain of 46dB + 8 steps programmable gain with step size of 3dB was applied to each sensor signal. The signal conditioning stage also has band pass filter to remove out of band noise and it was realised using a Butterworth 4^{th} order high-pass (25 KHz) filter followed by another Butterworth 4^{th} order low-pass (85 KHz) filter. The system has been designed to sample 512 analog inputs, simultaneously at 196 KHz and transfer the digitized data to the receiver over the fiber optic link. (Even though we had only 508 sensors, the hardware is capable of acquiring data from 512 channels by virtue of its design).

The energy efficient design (to keep the power consumption low) demanded minimal quantity of complex hardware components in the wet-end electronics. The design requirement was satisfied by a 64-channel data acquisition (DAQ) card *(PMC66-16AI64SSC)* from General Standards Corporation. This *PC104Plus* DAQ module is capable of sampling 64 channels simultaneously up to 200 KHz per channel with each sample being 16 bit wide. This module is connected to a PC104plus host processor board for data transfer to the receiver onboard. The architectural overview of ANI wet-end system is shown in Fig. 2.

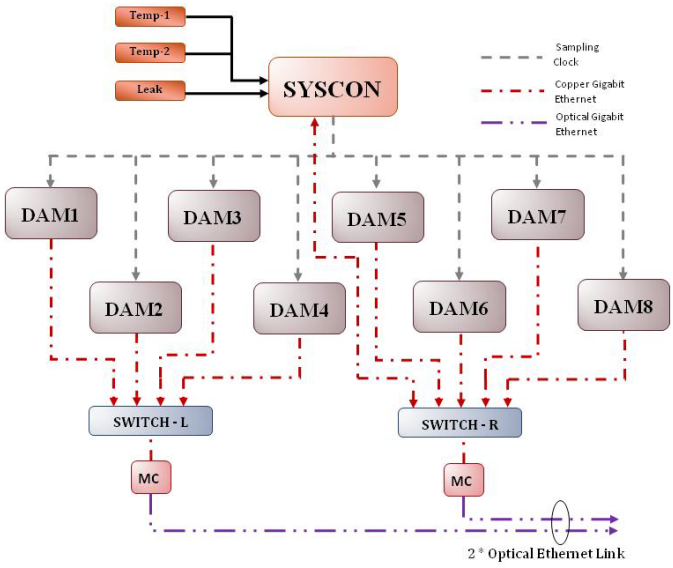

Figure 2. ANI architectural overview

There were 8 Data Acquisition Modules (DAM), with each DAM consisting of a DAQ card and a CPU board. The CPU board is an Intel Atom based (1.6GHz) PC104plus form factor embedded single board computer (SBC). This SBC also has onboard gigabit Ethernet NIC, capable of transferring data at a maximum rate of 1000 M bits/sec. The SBC runs on *"tinycore-linux"* operating system, a light weight Linux distribution. The complete wet-end system was monitored and synchronized by the System Controller (SYSCON). Two gigabit Ethernet switches were used to multiplex the data from the 8 DAMs, where each switch connected 4 DAMs to the receiver onboard. The Media converter (MC) at the wet-end side was able to extend a copper based Gigabit network via optical fiber cable which in our case was 160 meters.

A. *SYSCON*

The overall system had one master controller, which can control data acquisition, front-end gain, and the power supply to all the modules. It also monitored the temperature inside the wet end casing and also for any water leakage in the system. Fig. 3 shows functional diagram of SYSCON.

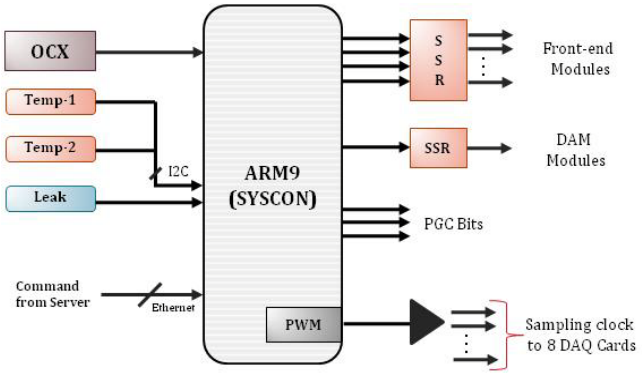

Figure 3. SYSCON Functional Diagram

The SYSCON was built around an ARM based processor, STR912. The receiver communicated with SYSCON via one of the optical Ethernet link. The wet-end system was equipped with 3 environmental monitoring sensors, 2 temperature sensors interfaced through the I2C bus and one leak sensor connected via GPIO. The OCX (oven controlled oscillator) provided a low drift external clock for generating the sampling clock for the DAQ. The SYSCON generated the actual sampling clock using the OCX and its internal pulse width modulator (PWM). Three GPIO pins were programmed to provide the 3-bits for front-end Programmable Gain Control (PGC). The SYSCON also controlled the power supply to DAM and sensor modules through solid state relays (SSR). These various functions performed by SYSCON were all achieved through commands sent over Ethernet from the dry-end system by the operator.

B. *Data Acquisition Module*

The DAQ card supported external trigger to sample the signals and this was utilised for sampling all 64 channels simultaneously. All 8 DAQ cards were synchronised by connecting their sampling clocks to one common source (SYSCON). The functional diagram of Data Acquisition module is shown in Fig. 4. The sampling clock triggered the *"Sample & Hold"* circuitry to hold the 64 analog voltages simultaneously. Then the 8 high-speed ADCs converted them to digital sequentially. The DAQ filled its internal FIFO with 196 Mbits of data per second during acquisition. This was

well below the PCI bandwidth *(132 Mbytes/sec)*. The SBC sets the DAQ with threshold number bytes required to initiate the data transfer to SBC. Once the FIFO is filled with more than this threshold data, it notified the SBC. The SBC then initiated a DMA transfer to its internal memory.

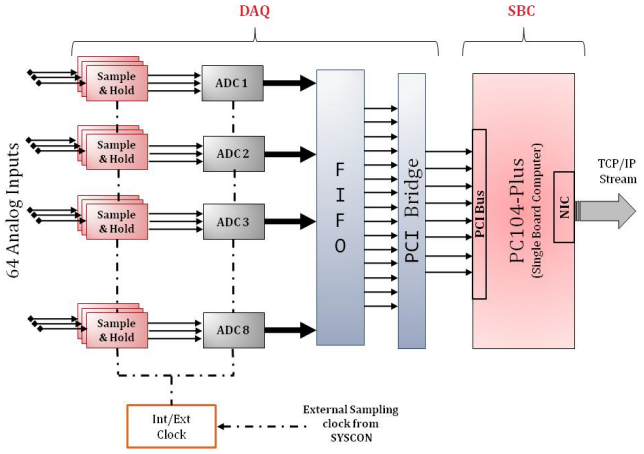

Figure 4. DAM functional Diagram

The real-time data transfer from the DAQ required a real-time operating system or a very light weight non real-time OS. The light weight Linux distribution *"tinycore-linux"* running on the Intel Atom CPU was able to transfer the real-time data to internal memory. A highly efficient way to transfer this data to the surface system was over Gigabit Ethernet. The SBC has an onboard Ethernet Network Interface Controller *(NIC)*. Using the connection oriented TCP/IP protocol the data was transmitted to the dry-end system without packet loss. The efficient implementation of TCP/IP in Linux supported the deployment of Linux-based operating systems on both wet and dry end systems.

C. *Data Transmission*

Two commercial Gigabit Ethernet switches were used to connect all the DAM modules and send the multiplexed data to the server. The dry-end server and wet-end system were connected by a 160 meter cable. The requirement of driving high speed data over the long distance *(160 meter)*, was satisfied by two RJ45 to SFP media converters to convert the *copper Ethernet* to *optical Gigabit Ethernet* link at the wet-end. This optical link is converted back to copper link by two more media converters at the dry-end side.

D. *Auxiliary Sensors*

Auxiliary sensors were required to ensure smooth and safe operation of system in the field. The system was equipped with two temperature sensors and one leak detection sensor for safety purposes. The outputs of these sensors were monitored via SYSCON through queries. The water leakage will force SYSCON to shutdown power to all modules. If the temperature reaches near the maximum operating range specified for the hardware, the SYSCON will alert the operator and the operator can decide whether to continue with the acquisition.

E. *Power Supply Design*

One of the challenges we faced during power supply design was the management inrush current when the complete system was switched on. There were 508 sensor modules and associated electronics in the system, and switching all modules ON at the same time would cause a high instantaneous input current. A solution to address this problem was to divide the total load into smaller sections and turning the units on one by one. Fig. 5 shows the typical power supply distribution scheme we employed.

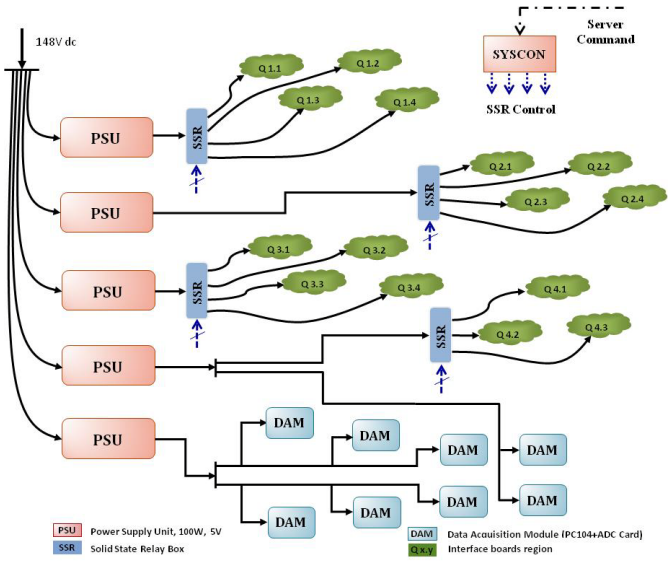

Figure 5. Power distribution scheme

The total front-end electronics load was divided into four quadrants and each quadrant was equipped with a dedicated 100 W power supply unit (PSU). Each quadrant was further divided into 4 sub-quadrants. The sub-quadrants were controlled by four solid state relays (SSR). These SSRs were controlled by SYSCON with commands from the dry-end server to turn them on one after the other. This approach activated only 36 sensor modules powered by one power supply at any one time. To optimize power consumption the SYSCON powered up the sensor modules only for the period of acquisition, which was decided by the schedule of acquisition.

F. *Programmable Gain*

Programmable gain control (PGC) is an important feature in any underwater acoustic system. The strength of the signal varies depending upon the sea environment. A fixed front end gain will not help to utilize the full dynamic range of the ADC. For example setting a high gain may lead to ADC bits saturation if operated in high acoustic activity area. Similarly a very low gain may not utilise the full capability of the ADC in a low acoustic activity area. The 3-bit control allowed up to

8 different gain setting. PGC control bits were generated at the SYSCON and the same gain was applied across all sensor modules. Fig. 6 shows the PGC wiring strategy.

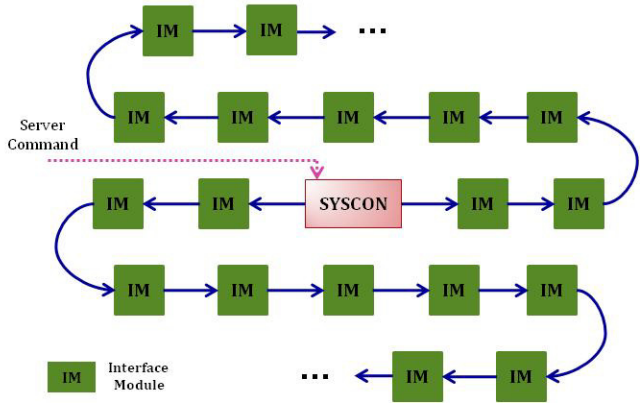

Figure 6. PGC wiring strategy

The sensor modules were connected to an interface module (IM), which had provision for connecting up to 10 of them. The SYSCON supplied PGC bits to the nearest interface module. The bits were cascaded and buffered through IM to provide PGC bits to other sensor modules. This approach effectively propagated PGC bits to entire sensor modules with less cable routing. This approach also helped in locating the faulty IM board *(by tracing PGC)* during assembly. The introduction of PGC provided some operational flexibility when operated in different environments compared to the previous system.

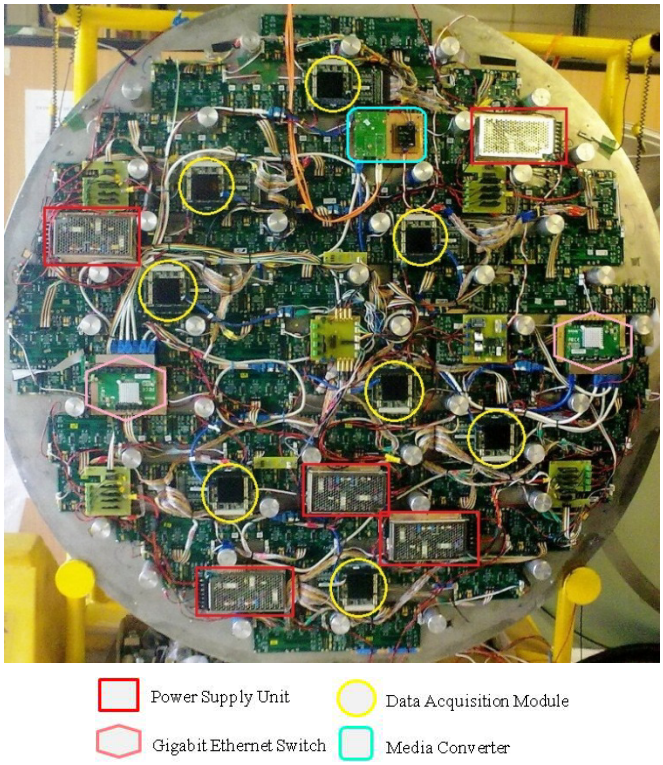

Figure 7. Array picture captured during assembly

The actual picture of the array electronics captured during the system assembly is shown in Fig. 7.

G. *Dry-end system*

The primary functions of the dry-end system were to receive the acoustic data from 8 wet-end computers *(DAM)*, and command/control the entire wet-end system through SYSCON. The system at the dry-end should be fast enough to receive streams of TCP/IP packets from wet-end computers. The dry-end receiver system was an industrial *"tower-server"* computer, powered by 2 Intel Nehalem Processors with two onboard Gigabit Ethernet network controllers.

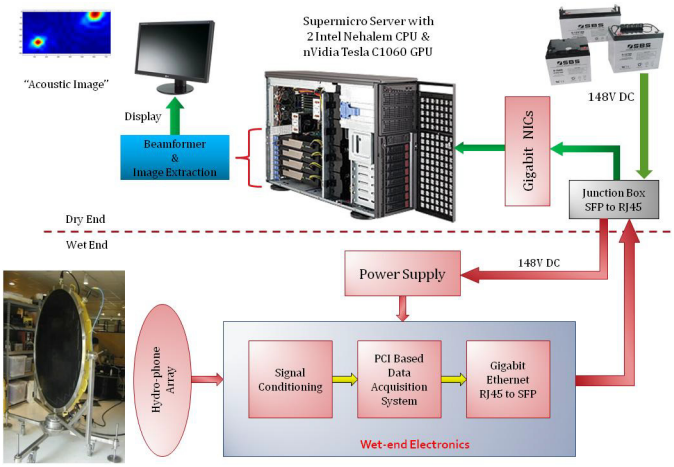

Figure 8. End-to-End system block diagram

The storage media used was a RAID-0 array of four 1GB SATA hard-disks. The RAID-0 configuration was about 4 times faster (as compared to a single HDD) in hard-disk writing speed due to the use of 4 disks. Fig. 8 shows the block diagram of the complete system. The dry-end server is also equipped with 2 NVIDIA – Tesla C1060 PCIe GPGPU (General Purpose Graphics Processing Unit) cards for signal processing. Each card delivered up to 1 TFLOPS of computing power. The GPU provided critical assistance in processing the field data.

This is a major improvement from the previous system in terms of both cost and complexity. The earlier system with fibre channel technology required 4 Raid-0 arrays of 14 fibre channel hard disks each to cater for the 54 embedded processors. The four raid arrays were controlled by four PC platforms for receiving and writing the data, all in a synchronised manner. The processing was carried out on a PC cluster which was about 5 times costlier and bulkier.

III. Software Architecture

The software architecture used a simple nested producer-consumer model. There were 8 producers (DAMs or clients) and one consumer (dry-end server). Each producer produced

data at the rate of 196 Mbits per second so that the consumer received an overall data rate of 1.6 GBits per second.

In addition to accepting sensor data from the embedded clients, the dry-end server also acted as a net-boot server for each client. This allowed the clients to obtain an OS image from the server at boot-time itself, thus eliminating the need of an onboard flash disk/HDD for booting an OS (In the earlier system each embedded system had the OS stored in a compact flash disk attached to each of them at times causing reliability issues while booting up). The *"tinycore-linux"* distribution client OS supported this net-boot feature. Each of the 8 clients received an IP address from a DHCP server program (running on the server) at boot-up. The clients were controlled from the server using remote shell protocols such as SSH (secure shell). The server application software (which waits for and handles client connections) was launched prior to remotely launching the client acquisition application software.

A. Server Architecture

The server's main job was to read data from the clients. It checked for integrity of data before writing them to the storage disk. The server-clients used TCP/IP protocol for reliable data transfer. Even though TCP/IP implementation on Linux was very efficient, the requirement of server processing power was also very high, as it needed to communicate with eight clients concurrently.

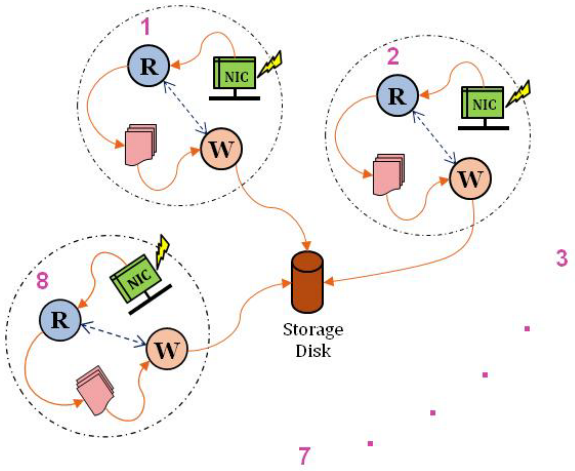

Figure 9. Server software architecture

The server was powered by two quad-core processor with Intel Hyper Threading technology allowing up to sixteen threads to run concurrently. The data acquisition process was initiated by the server which was connected to 8 clients and the ARM-based system-controller (SYSCON). The server-side application gathered the acquisition parameters such as the period of acquisition, programmable gain values, etc from the user and waits for client-side applications to request for a connection. The server then created eight processes upon request from the clients and sets up the acquisition on the clients. The server software architecture and flow diagram are shown in Fig. 9 and 10 respectively.

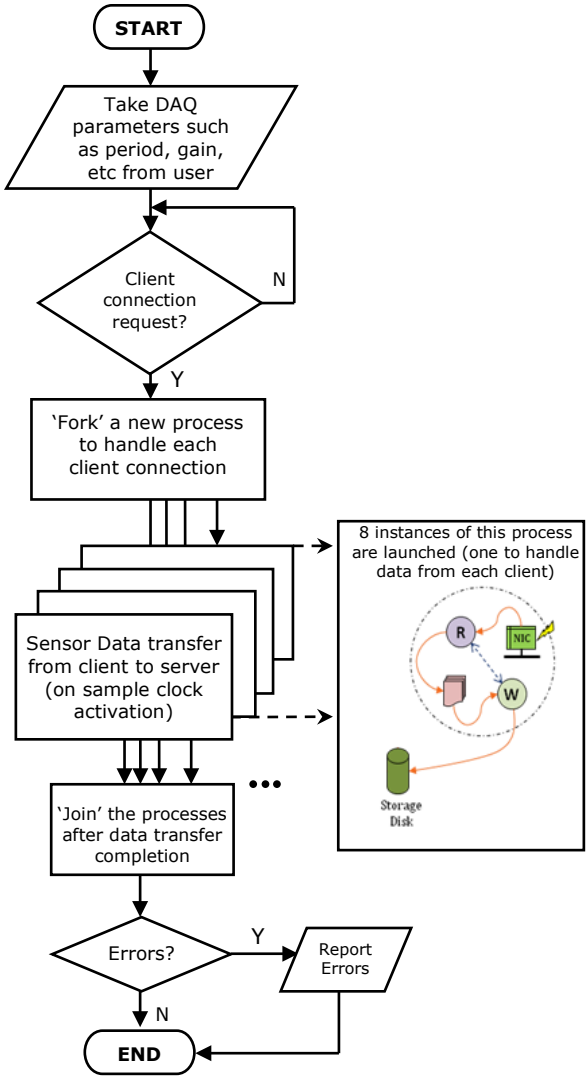

Figure 10. Server software flow diagram

Once the acquisitions begin, each process created two threads, a *read* thread (producer) and a *write* thread (consumer). The producer then reads the data from the Network Interface Controller (NIC) to a linked-list array of buffers in the main memory (RAM). The consumer reads the data from this buffer-array and writes to files on the disk-storage.

B. Client Architecture

On the server's request each client ran a data acquisition application that transferred data from its 64-channel ADC card (on its PCI bus) to the server. When the SYSCON was requested by the user (at the server) to enable the sampling clock, all the ADC cards acquire data and fill their respective FIFOs. Each client's job was then to read the data from its ADC card's FIFO to its main memory (RAM) and write it to NIC for transmission over Ethernet (TCP/IP) to the server. The client software architecture and flow diagram are shown in Fig. 11 and 12 respectively.

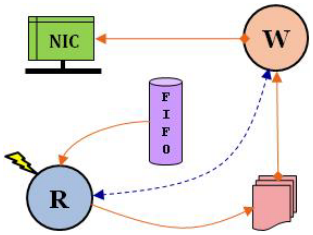

Figure 11. Client software architecture

The client software architecture is also a producer-consumer model similar to the server software. The producer thread reads the data from the ADC card and fills in to the double buffer in the main memory.

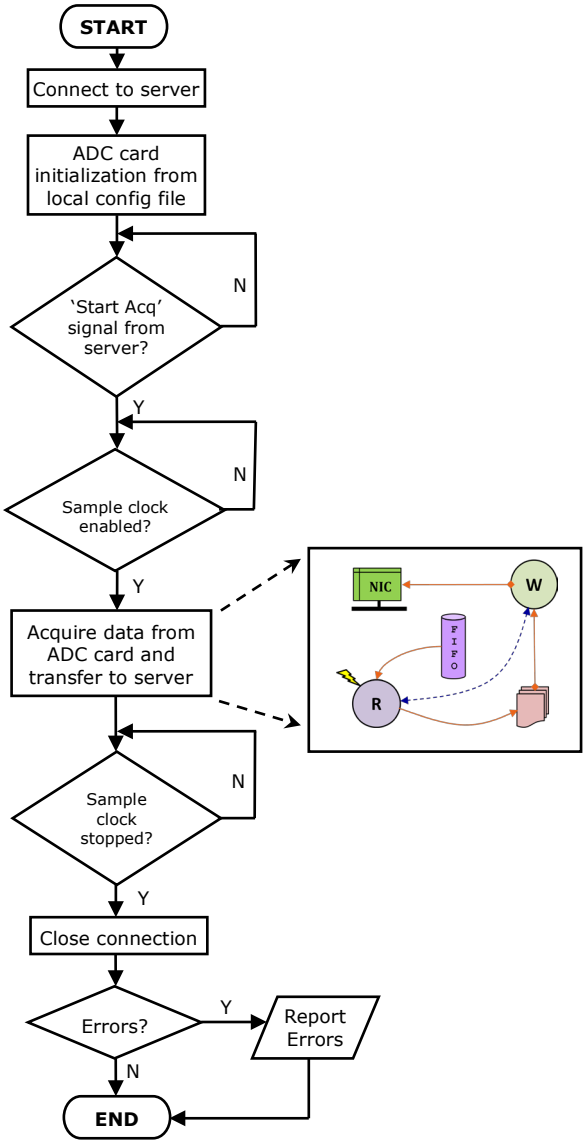

Figure 12. Clients software flow diagram

The consumer thread reads data from a linked-list array of buffers, writes it to NIC and it also handles the retransmission. Since the read and write operation used DMA engines to transfer data, multithreading was provided for real-time performance of the data transfer. In order to optimize power consumption, the clients switched over to a slower clock-speed (800 MHz) power-saving mode, when not acquiring data. When the processing load increased during acquisition they switch back to a performance mode (1.6 GHz).

IV. Noise Reduction

One of the most challenging aspects of designing any underwater data acquisition system was to have the sensor signals free from in-band noise. Since the band of interest was very large (25 KHz to 85 KHz), noise from various sources could intrude in to the operating bandwidth. For an Imaging system, the in-band noise will appear as a target pixel at the broadside, because it will be coherent across all the sensor data. The various noise sources are switching noise from the dc-dc converters, thermal noise, electronic noise and electromagnetic interference (EMI). Amongst these switching noises from the dc-dc converters were found to be the most dominant one in our design. Selection of switched mode power supplies played a vital role in a low noise design. In our application we found that high power small form-factor dc-dc converters were the major contributors of noise. These converters were switched at very high frequencies to minimise the size. Fig. 13 is a spectral plot of one of the acoustic channels when powered by a high frequency compact dc-dc converter. Switching noise (picked up through conduction and radiation) from these tiny modules generated multiple in-band tonals. Several filtering attempts were made to remove these tonals, but without much success.

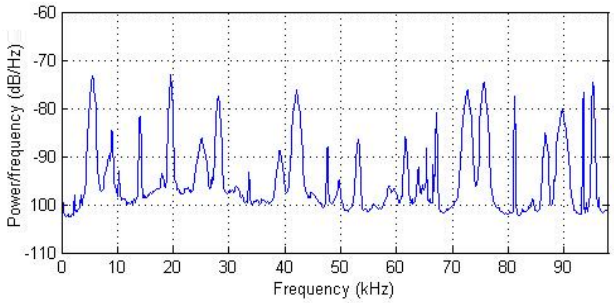

Figure 13. Effect of HF compact dc-dc converter

In our final design the dc-dc converters selected were such that the 3rd harmonic of their switching frequency was well below the lower cut off frequency (25 KHz) to keep the in-band noise as low as possible. Fig. 14 is a spectral plot of the same acoustic channel given before when powered by the new dc-dc converter. Although these converters were bulky, they were less noisy and very effective for high frequency broad band underwater systems. The complete wet-end system was designed to work with single voltage. The single and low voltage design *(5V)* eliminated cascaded converters, which greatly reduced electronic noise in the system. The system was powered from lead-acid batteries to avoid any conducted Electromagnetic Interference generated by alternative power

sources such as a generator or inverter. This was made possible mainly by the reduction in power consumption by about 70% from its former counterpart.

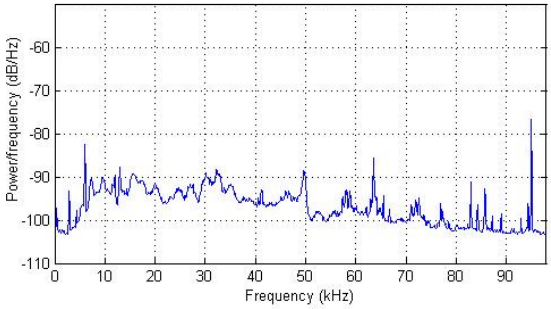

Figure 14. Effect of LF dc-dc converter

The current design consumes a power of only 450W as against the 1700W of the earlier system.

V. Performance Evaluation

The redesigned system was deployed in Singapore waters for ambient noise imaging studies. This provided the best opportunity to test the endurance and performance of the data acquisition system. Over 14 days, almost 2 terra bytes of high quality data were collected and remarkable Images were formed using this data. A sample image of an acoustic ping at 120m distance is shown in Fig. 15.

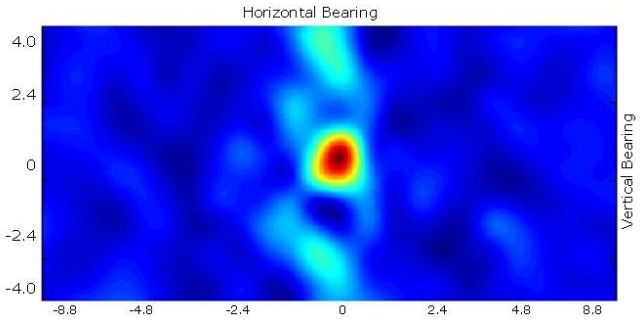

Figure 15. Image of an Acoustics ping at 120m

VII. Achievements Summery

The re-design of the data acquisition system has resulted in a power consumption of about 450W in place of 1700W by the earlier system with a substantial power saving of over 75%. This resulted in the reduction of heat generated internally and improved the endurance of the system. The big savings in power also resulted in a battery operated data acquisition system freeing up many noise problems arising from the generators that were used to power up the earlier system. The data collected using the new equipment was found to be clean and less contaminated by such noises. Previous design did not have any provision for changing the gain of the front-end analog system. In the current design the user can select eight different gain settings in the field and also set them remotely. This provided operational flexibility to combat different acoustic environments, specifically when operating with active sources, and for maximising the dynamic range. The use of a more robust Linux operating system, as compared to the previous Windows NTe, on the clients has increased the reliability of the system for repeated data collection. The redesign also resulted in a more compact and less costly dry-end receiver and processing system. A single high end server with a few terabytes of hard disk in a RAID- 0 format provided the data storage solution in place of 56 fibre channel hard drives and four PCs used earlier.

Conclusion

A novel architecture for an underwater imaging array has been developed with improved reliability, higher endurance and bare minimum power consumption at a lower cost compared to its previous counterpart. The design, development and field trial results have been described in this paper. Hierarchical Noise reduction approaches were used to eliminate the in-band Noise. The well known protocol mode of data transfer (TCP/IP) made the onboard electronics expandable and reconfigurable in minimal time. Also it was proven that Gigabit Ethernet with fiber extension can even transmit up to 80% of its capacity over hundreds of meters without any special requirements. The system, though developed for ANI in underwater, may find use in other applications that require large volumes of data to be collected and transported to long distances over a fiber optic link. To the best of our knowledge the design is unique and probably the first of its kind developed for underwater applications.

Acknowledgment

The Authors would like to express their gratitude to their colleagues Asst. Prof. Mandar, Mr. Naveen, Ms. Soo Pieng, Mr. Manu and Mr. Sagar from ARL for providing constant support to the system development; Mr. Unnikrishnan, Mr. Jaiwin and Mr. Mohanan for their support in the field operations. Acknowledgments are also due to the Defence Research & Technology Office (DRTech), Singapore for supporting Ambient Noise Imaging studies for the past decade in ARL.

References

[1] V. Pallayil, M. A. Chitre, E. T. Tan, J. R. Potter, T. B. Koay, S. B. Ruiz, and S. P. Tan, "Ambient noise imaging - First deployments of ROMANIS and preliminary data analysis," in OCEANS MTS/IEEE Conference on Celebrating the Past - Teaming Toward the Future, San Diego, CA, September 22-26, 2003

[2] V. Pallayil, P. D. Deshpande, S. Badiu, S. Constantin, B. Lu, and J. R. Potter, *"A 1.6 Gigabit/second, 25-85 kHz Acoustic Imaging Array- Novel Mechanical and Electronics Design Aspects," in* Oceans 99 MTS/IEEE Conference Proceedings, Seattle, vol. 1, September 1999

[3] H. Zhang, T. B. Koay, V. Pallayil, Y. Zhang, and J. R. Potter, "Fibre channel storage area network design for an acoustic camera system with 1.6 GBits/s bandwidth," in *IEEE REGION 10 International conference on electrical and electronic technology* (D. Tien and Y. Liang, eds.), vol. 1, 2, pp. 143-148, August 2001.