



**Michigan  
Technological  
University**

Michigan Technological University  
**Digital Commons @ Michigan Tech**

---

Dissertations, Master's Theses and Master's Reports

---

2017

# A TESTBED DESIGN FOR MONITORING THE LONG-TERM SPATIAL-TEMPORAL DYNAMICS OF UNDERWATER ACOUSTIC CHANNELS

Krishna Chaitanya Poduru  
*Michigan Technological University, kcpoduru@mtu.edu*

Copyright 2017 Krishna Chaitanya Poduru

---

## Recommended Citation

Poduru, Krishna Chaitanya, "A TESTBED DESIGN FOR MONITORING THE LONG-TERM SPATIAL-TEMPORAL DYNAMICS OF UNDERWATER ACOUSTIC CHANNELS", Open Access Master's Report, Michigan Technological University, 2017.  
<https://digitalcommons.mtu.edu/etdr/535>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etdr>



Part of the [Computer and Systems Architecture Commons](#), and the [Electrical and Electronics Commons](#)

A TESTBED DESIGN FOR MONITORING THE LONG-TERM SPATIAL-TEMPORAL  
DYNAMICS OF UNDERWATER ACOUSTIC CHANNELS

By  
Krishna Chaitanya Poduru

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

© 2017 Krishna Chaitanya Poduru

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Report Advisor: *Dr. Zhaohui Wang*

Committee Member: *Dr. Glen Archer*

Committee Member: *Dr. Jeremy Bos*

Department Chair: *Dr. Daniel R. Fuhrmann*

# Contents

1.	INTRODUCTION.....	1
1.1.	Motivation for this work .....	1
1.2.	Testbed design aspects and considerations.....	2
1.3.	Brief description of the overall system .....	3
1.4.	Existing work .....	4
1.5.	Differences between SEANet G2 design and current system design.....	5
2.	SINGLE NODE DESIGN .....	6
2.1.	Single node architecture .....	6
2.2.	Functionality of the individual components .....	8
2.2.1.	Computing subsystem .....	8
2.2.2.	Communication subsystem .....	9
2.2.3.	Battery subsystem.....	18
2.2.4.	Voltage regulation subsystem .....	19
2.3.	Single Node System operation .....	20
3.	TESTBED ARCHITECTURE FOR MONITORING THE SPATIAL-TEMPORAL DYNAMICS OF UNDERWATER ACOUSTIC CHANNELS .....	22
3.1.	Overview of the testbed architecture .....	22
3.2.	Server level architecture and algorithm.....	23
3.3.	Stability and error handling mechanisms.....	24
3.4.	Interfacing MySQL database with MATLAB.....	25
4.	GUI DEVELOPMENT .....	26
4.1.	Introduction and architecture.....	26
4.2.	GUI at server level .....	27
4.3.	Web GUI .....	27
5.	A FIELD EXPERIMENT IN THE PORTAGE LAKE .....	28
5.1.	Experiment overview .....	28
5.2.	Experiment procedure .....	29
5.3.	Results .....	30
6.	KEWEENAW SPATIAL-TEMPORAL EXPERIMENT .....	31
6.1.	Experiment overview and setup.....	31
6.2.	Results .....	31
6.3.	Problems faced and solutions .....	33
7.	CONCLUSIONS AND FUTURE WORK .....	34
8.	REFERENCES.....	35

## List of figures

Figure 1. 1 System architecture.....	4
Figure 2. 1 Single node architecture .....	6
Figure 2. 2 Single node testbed system.....	7
Figure 2. 3 Raspberry pi.....	8
Figure 2. 4 Acoustic modem.....	10
Figure 2. 5 Acoustic modem connector .....	10
Figure 2. 6 Acoustic modem connector pin .....	11
Figure 2. 7 RF Modem.....	13
Figure 2. 8 Keweenaw Waterway Test site.....	17
Figure 2. 9 Portage Lake Test Site.....	18
Figure 2. 10 Battery pack.....	19
Figure 2. 11 DC-DC converters .....	19
Figure 2. 12 single node system operation.....	21
Figure 3. 1 High level view of server and node connections .....	22
Figure 3. 2 Server level architecture and algorithm.....	24
Figure 4. 1 GUI Architecture .....	26
Figure 4. 2 Web GUI .....	27
Figure 5. 1 Experiment overview.....	28
Figure 5. 2 (June 2014).....	28
Figure 5. 3 (March 2016).....	29
Figure 5. 4 Results of portage lake experiment [courtesy of Wensheng Sun].....	30
Figure 6. 1 Experiment setup.....	31
Figure 6. 2 Keweenaw spatial-temporal experiment [courtesy of Wensheng Sun].....	33

## List of tables

Table 2. 1 Connector pin details .....	11
Table 2. 2 Acoustic modem serial settings .....	11
Table 2. 3 RF Modem Specifications.....	13
Table 2. 4 Default on state settings.....	14
Table 2. 5 Serial Baud Rate settings .....	15
Table 2. 6 Wireless Link Rate.....	15
Table 2. 7 Output Power settings .....	15
Table 2. 8 RF Modem RSSI Values at Keweenaw Waterway Test site .....	17
Table 2. 9 RF Modem RSSI Values at Portage Lake Test site .....	18

## Acknowledgements

First, I would like to thank my advisor Dr. Zhaohui Wang for guiding me so patiently even though I made many mistakes and delayed the work so long. Apart from the research and her subjects, she used to guide me every subject I have taken at tech. Apart from the technical information, one other great thing I have learned is her hard work. She always sets an example of it.

I would like to thank Mr. Wensheng sun, Mr. Li Wei, and Mr. Chaofeng Wang. Along with my advisor, they are always there for me. They helped me a lot in every step of my research and making this report. I learned many simple tips from them which made my life more easier. They showed me how professional an engineer must work in his research and also during experimentation.

Last but not least, I would like to thank my entire UWSN team for making this journey a wonderful experience. I loved every part of every experiment, the design, planning, execution and analysis.

## Abstract

The underwater acoustic network testbed helps to validate the theoretical results and bridge the gap between experimental results. Characterizing and modeling the spatial-temporal dynamics of underwater acoustic channels is essential to developing efficient and effective physical-layer communication algorithms and network protocols. This work dedicates to designing a testbed system to measure the spatial-temporal dynamics of underwater acoustic channels. The collected measurements will shed insights into the spatial-temporal correlation of underwater acoustic channels and will be used to evaluate the theoretical algorithms that are designed to model the spatial-temporal dynamics and to exploit the spatial-temporal dynamics for more efficient and effective underwater system operations.

The report speaks about how to tackle the above problem and discusses the following aspects in detail which are, individual node design which is controlled by a raspberry pi, comparison of the current test bed with the existing testbeds in field, complete description of the server algorithm and its error handling techniques, development of the server level GUI and web-based GUI and finally some of the experimentations carried out in Portage lake and Keweenaw bay.



# 1. INTRODUCTION

This chapter is about the motivation of this work, the testbed design considerations, and overall testbed architecture.

## 1.1. Motivation for this work

Underwater acoustic communications enable voice or data telemetry between sensor nodes which are deployed underwater. They have many applications <sup>[15]</sup> which are addressed below.

1. It assists navigation by identifying the hazards on the seabed based on the position of dangerous rocks.
2. It helps in disaster prevention by measuring the seismic activity from remote locations which helps to find tsunamis.
3. It helps in underwater explorations such as detecting the oilfields, routes for laying undersea cables and helps in finding valuable minerals.
4. It helps in tactical surveillance. This sensor data is used by submarine and small delivery vehicles.

Some of the problems<sup>[15-19]</sup> which are associated with underwater acoustic channels are addressed below.

- First, the underwater acoustic channel is very bandlimited due to the significant attenuation of high-frequency acoustic signals in water. Usual frequencies which are associated with the underwater communications are between 10 Hz and 1 MHz depending upon the application. The main reason for sound attenuation at higher frequencies in sea water is due to the viscosity<sup>[19]</sup>.
- Secondly, the sound wave travels in multiple paths and with huge time delays, often covering hundreds of symbols which cause Inter-Symbol Interference. The direction of sound propagation is dependent upon the sound speed gradient in the water. Inside the sea usually, the vertical gradients are greater than the horizontal gradients. Along with this factor and many obstacles create the multipath travel of sound.
- Third, due to different changes in wave characteristics (phase, amplitude) when hitting with different reflectors such as sea-air boundary line, seabed or any other obstacles causing destructive interference of the wave due to multipath travel leads to fading.

- Fourth, the continuous time-varying nature of the channel and many noise sources such as ocean turbulence and microseisms at lower frequencies. Wind-induced surface noise is the primary source between 1 kHz and 30 kHz. The thermal noise of water molecules is a reason at higher frequencies. All these factors make it even more tough to track the medium properties.

The system performance in the Under Water Acoustic Networks (UWAN's) depends upon the location of the network components and the time of transmission jointly<sup>[12]</sup>. A significant challenge in UWAN<sup>[9, 10]</sup> is to develop the medium access control, routing and physical layer protocols due to the significant propagation delay caused by the low speed of sound<sup>[8]</sup> under water. There are several terms which are essential and are addressed below.

- **Temporal dynamics:** The characteristics of underwater channels varies continuously with respect to time due to several factors.
- **Spatial dynamics:** Keeping the distance between two nodes constant, the channel properties between the two nodes varies with their geographical location.
- **Fast fading:** Usually the reason for fast fading is due to the multipath propagation. For example, when signals travelling from multiple paths meet at the receiver, they might either constructively interfere or destructively interfere depending on the phase difference. Phase difference is directly related to the distances traveled by the signals. Even though in our current setup there is no mobile node, the surface dynamic change and mobile node movement due to waves can cause these issues.
- **Slow fading:** This is usually caused when there is a big obstacle in the way of the signal. In underwater acoustic communications, the general reason is a change in sound speed profile.

The following section speaks about the testbed design aspects and considerations and describes the feasibility of the platform to perform the experimentation<sup>[24]</sup>.

## 1.2. Testbed design aspects and considerations

To measure the spatial-temporal dynamics of underwater acoustic channels, multiple geographically distributed acoustic nodes need to be deployed. Where the acoustic channel between each pair of nodes can be probed through acoustic transmissions according to a pre-determined schedule. The deployment will typically last for a long-time duration, varying from a few hours to several days or even longer, to obtain sufficient insights about both the short-term and the long-term channel dynamics. Given the type of deployment, the testbed design needs to have the following properties.

**Reconfigurability:** The current testbed design enables us to reconfigure some of the attributes of the acoustic modem and RF modem even after we deploy the setup. One of the attributes we can tweak for the acoustic modem is its transmission power, and in case of the RF modem, wireless link rate and transmission power.

**Fault-tolerance:** One of the major attributes for the system is its stability. The current test bed is designed in such a way that even if any node fails to upload the data or uploads any faulty data, the server does not crash. Instead it puts zeros inside the database, thus maintaining its stability for continuous operation.

**Real-time operation:** The testbed helps in monitoring some essential attributes of the system like the remaining battery power and remaining memory in the acoustic modem, which helps to make better decisions about the situation. The second reason is by seeing the real-time data we get more intuition of the situation. Because we cannot record everything that is happening around us, we need to know some things instantly. In real time by just looking at the data and environment, we get more insight into the situation and also convenient.

**Energy-efficient operation:** Testbed design helps control several aspects such as acoustic transmission duration, number of messages which are to be transmitted. We can reduce or increase them depending upon the energy leftover in the system. In the future, we can control a number of radio transmissions by saving the data locally inside nodes.

Furthermore, we expect the testbed design is scalable in the number of nodes with just a few changes in certain levels of the system. The susceptibility of the system to crashing is handled gracefully (discussed in section 3.3) in the current design but had many places for improvement based on future requirements.

### 1.3. System description

The entire system is divided into three major subsystems. One is the node subsystem. The second is the server subsystem, and the third is the graphical user interface (GUI) subsystem. It is a centralized network which is address centric and controlled by the server. At the node level, there are four nodes communicating with each other using acoustic modems. The server subsystem dictates the communication sequence of the nodes and requests individual nodes for the acoustic channel data using RF links. It puts the required information of the acoustic channel in local a database. Then the GUI program from the GUI subsystem plots the data and creates number of functionalities that are used to control the system. The web-based GUI from GUI subsystem takes the Java Script Object Notation (JSON) objects created by the server GUI, which are continuously updated in real time and plot them on a web page when the appropriate button is clicked. The web-based GUI is deployed on an Apache server running alongside in server subsystem. This method of design is primarily based on the application requirement and previous system designs. Major components of each sub systems in the below Figure 1.1.

**NODES:** Raspberry pi, Acoustic Modem, RF Modem

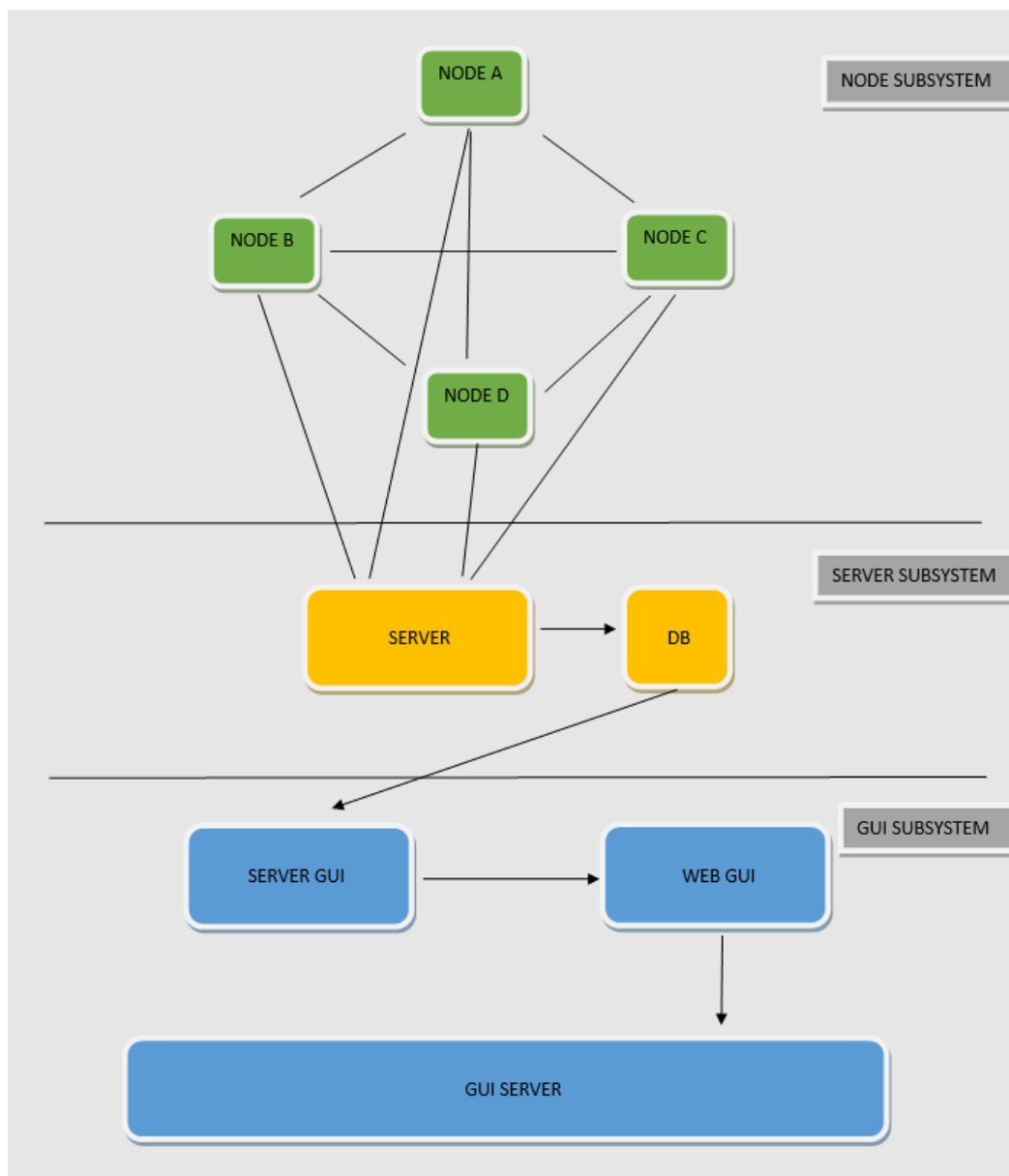
**SERVER:** MATLAB Program

**DB:** MySQL database

**SERVER GUI:** MATLABGUI program, JAVA GUI Program

**WEB GUI:** P5 Js, Javascript.

**GUI SERVER:** Apache server



**Figure 1. 1 System architecture**

#### 1.4. Existing work

Of many works on going in this field of research. SEANet G2<sup>[3]</sup> is a new high data rate software-defined underwater acoustic networking platform. This platform has achieved data rates on the order of 522 kbit/s over short links (10 m) with a bit error rate lower than  $10^{-3}$ . This platform is

also able to switch operational frequency at the run time with software defined adaption. SEANet G2 can run internet based applications and networking monitoring tools. This platform has the capability of harvesting power to recharge batteries from acoustic links.

The second existing work is Aqua-TUNE<sup>[1]</sup>, a testbed for underwater networks which it has been developed to efficiently deploy underwater networks and evaluate the performance of protocols and applications. The present testbed described in this report is inspired from Aqua-TUNE. Aqua-TUNE consists of similar high-level architecture associated with few differences. One difference is battery pack consisting of photovoltaic panels and its associated hardware. The second difference is this system uses two acoustic modems and the third difference is the Gumstix<sup>[5]</sup> controller which consume very high power compared to the existing controller used in our testbed

The third existing work is ORTUN<sup>[21]</sup>. Open Research Testbed for Underwater Ad Hoc and Sensor Networks[ORTUN] is the first open testbed for underwater community. The testbed is a buoy-based system which is more mobile and suits different environments. This testbed has two types of nodes with different capabilities. The first kind comes with a fixed physical layer which has the capability to use WHOI<sup>[22]</sup> micro modem or ISI SNUSE<sup>[23]</sup> modem which are connected by serial port. This system uses Gumstix<sup>[26]</sup> controller for managing the network protocols on the physical layer. For every ten to fifteen nodes of these kind have three gateways which are equipped with RF Modem. The second type of node is more advanced. It supports algorithm or protocol implementation and acoustic data collection at all layers of the network.

### 1.5. Differences between SEANet G2 design and current system design

The current system is different in many aspects when compared to SEANet G2.

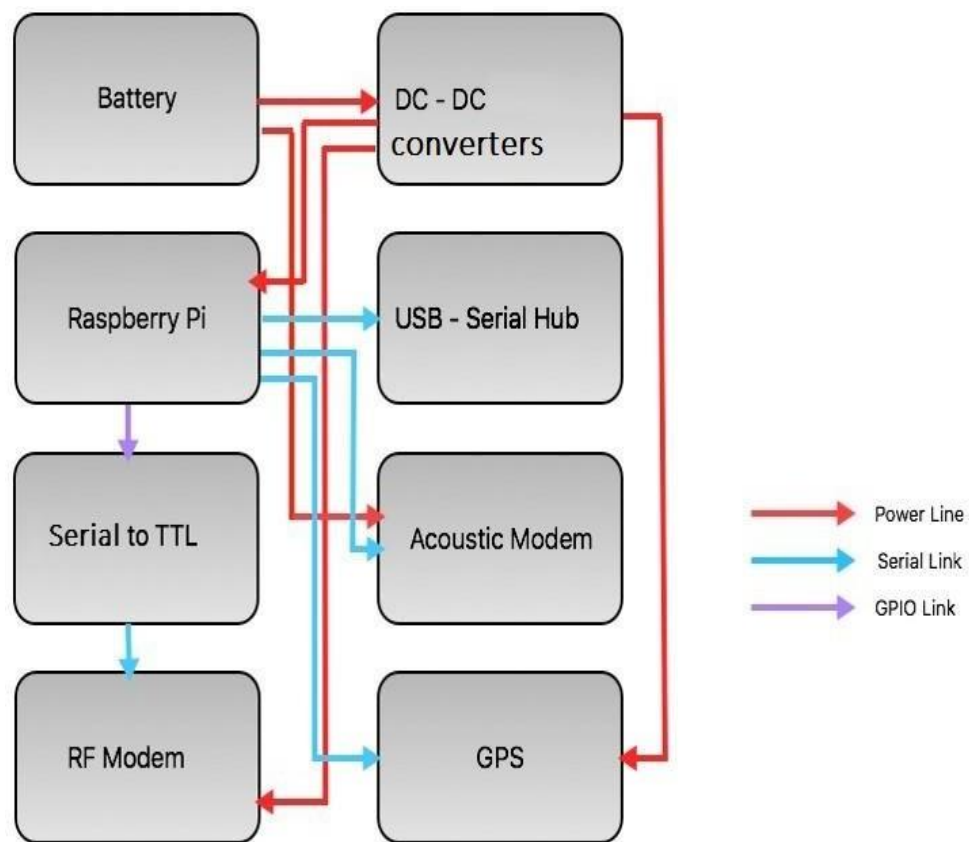
- SEANet G2 can achieve up to 522kbit/s over short links(10 m). The current design data rate is limited to acoustic modem speed which is 9kBit/s.<sup>[10]</sup>
- SEANet G2 can switch to different frequency of transmission during runtime. The current system's acoustic modem design supports only a single transmitting frequency.
- Both the system's physical layer have similar computing systems, SEANet G2 with Dual core ARM V9 processor and current system is Quad core ARM V8 processor. The current systems multi-tasking design capabilities are superior when compared to SEANet G2's. Because the current application running inside the node have high level of parallelism. One can exploit this possibility to divide the work on multiple cores.
- The cross layer controllable protocol stack capability of SEANet G2's is the same as to the current design. Both can have flexible software-defined networking stack.
- Both systems have a capability to run Internet based applications. But each system achieves that capability differently. The SEANet G2 platform hosts an adaption layer<sup>[11]</sup> that provides integration with the Linux IP Stack. Whereas the current design has this capability at the server level where the surface node can be controlled.

## 2. SINGLE NODE DESIGN

This chapter introduces the architecture of single nodes and describes the functionality of each individual component. Later parts discuss about system operations, signal flow and the power consumption of the testbed.

### 2.1. Single node architecture

The following section describes the single nodes architecture. The overall single node's architecture is displayed in the Figure 2.1.



**Figure 2.1 Single node architecture**

The major system components of the testbed are shown in the above picture Figure 2.1. They are the Battery, DC-DC Converters, Raspberry pi, USB-Serial Hub, USB to TTL converter, Acoustic modem, RF Modem and GPS. There are three major links describing the entire testbed, which are the Power line, Serial Link, and GPIO link. The power line shows the direction of the energy flow in the system. The serial link shows the data communication links in the system where as the GPIO link shows the interface between USB to TTL and Raspberry Pi.



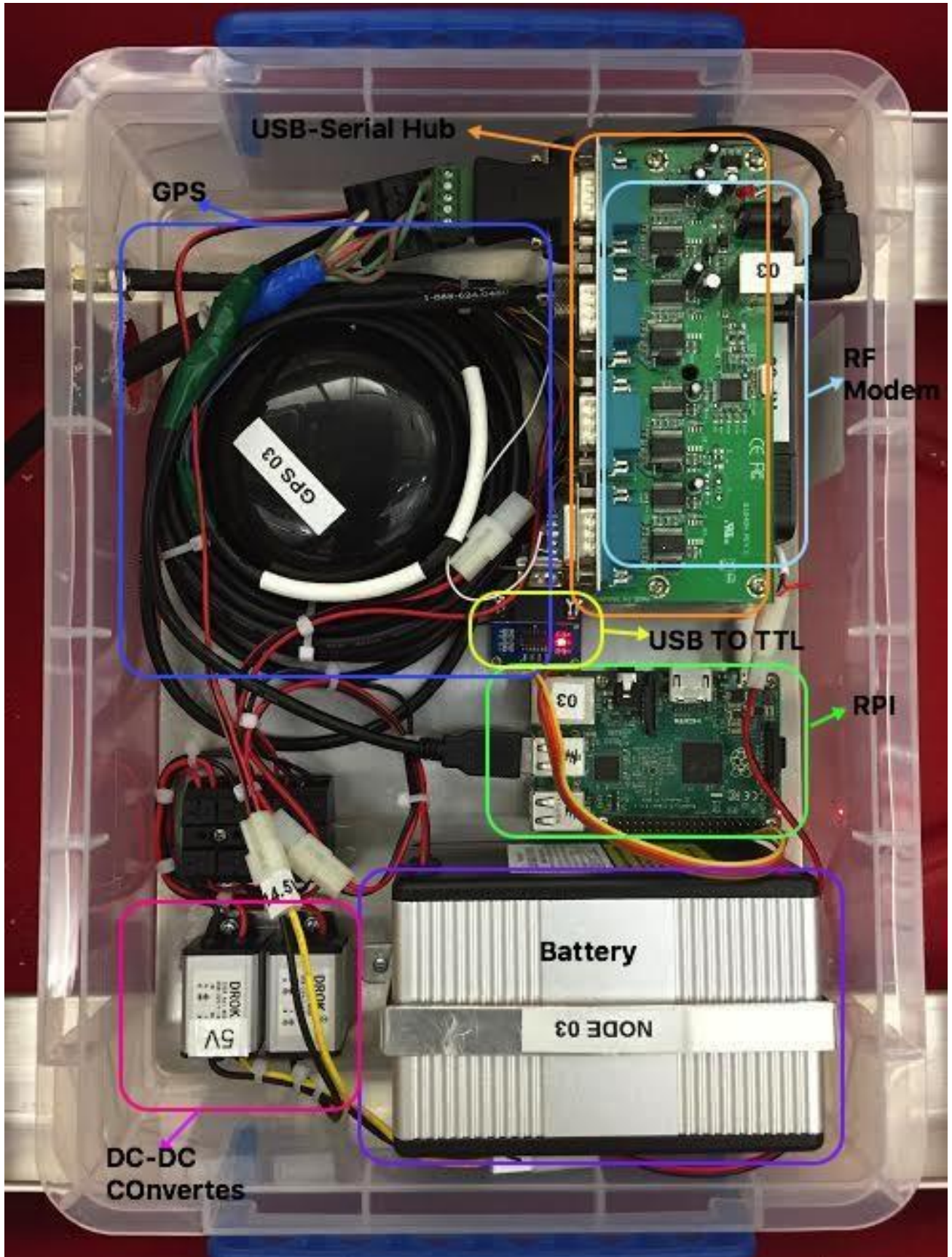


Figure 2. 2 Single node testbed system

## 2.2. Functionality of the individual components

The entire system can be classified into the following five sub systems, battery subsystem, voltage regulation subsystem, computing subsystem, communication subsystem and misc. This section describes the functionality of each subsystem and its components.

### 2.2.1. Computing subsystem

The main purpose of the computing system is to perform all the computations required for the respective node. It acts like a data acquisition module for different peripherals of the node. This subsystem consists of the raspberry pi 3<sup>[6]</sup> (RPI) model B shown in Figure 2.3 that is the computing module for each node. Some of its features of it are:

- ❑ A 1.2GHz 64-bit quad-core ARMv8 CPU
- ❑ 802.11n Wireless LAN
- ❑ Bluetooth 4.1
- ❑ Bluetooth Low Energy (BLE)
- ❑ 1GB RAM
- ❑ USB ports
- ❑ 40 GPIO pins
- ❑ Ethernet port

The operating system is raspbian. The protocols are written in Python.



**Figure 2. 3 Raspberry pi model B**

There are several other devices which are competitors to RPI which were not selected in the current system, Arduino has insufficient peripherals, Gumstix is another option with supporting peripherals and processing speed, but higher power consumption when compared to RPI is better. Intel Galileo Gen 2 has higher clock speed but some other characteristics was poorer. Raspberry pi is superior in terms of the resources available for development. Raspberry pi's OS Raspbian<sup>[12]</sup> is a light weight



distribution of Linux with wide range of compilers for Python, C, C++, JAVA etc. The Raspberry pi comes with JTAG <sup>[13]</sup> support which helps in debugging applications.

**Computing Operating system future scope:** There are so many distributions available for Raspberry pi which are compact, using a customized distributions yields more efficiency with less energy consumed. There are several reasons for this:

- General OS like Debian runs many processes in the back ground which consume more energy and are unnecessary for our application.
- CPU resources are wasted on many wrong processes as the UNIX kernel is based on a time sharing scheduling.
- Building our own distribution lets us to select only required features. We can also have our own energy management plans and Linux kernel debugger.
- We also can have a smaller system image which leaves more storage to save the files locally.

There are many new open source distribution managers such as Yocto<sup>[14]</sup> project, which has a very easy platform for creation of custom Linux distributions and includes board support packages for many existing ARM powered boards.

**Computing Hardware future scope:** For current application Raspberry pi has over computing capabilities. If the application remains constant and the number of nodes seems to be increasing then the next potential is ARM Cortex M processor series because these kind are highly energy efficient processors. Here are some more reasons why.

- Current application requires two RS 232 Serial ports, significant internal flash memory such as 16 MB, Wifi capabilities for future applications.
- For above mentioned features there are many potential options which are commercially available, one of which is TI cc3220 sf. The same controller is used in now a days many commercial IOT applications.

### 2.2.2. Communication subsystem

The main purpose of this communication subsystem is to interface with the outside world through air and water. The components of this system are the acoustic modem and RF modem.

#### **Acoustic modem**

This is the heart of every node. All the algorithms we run on the RPI or rest of the system are designed to control the acoustic modem remotely. The acoustic modem, shown in Figure 2.4, is designed and developed by Aquasent Technologies has a serial RS-232 interface. There is a specific set of commands used to control the modem. The modem can be thought of a state machine

which at every time can be described by a set of register values which can be configured using the specific set of commands.



**Figure 2. 4 Acoustic modem**

#### **Acoustic modem mechanical interface**

The modem connects to other systems through a single cable. Both power and data come through this cable. It has a wet pluggable connector that plugs to the modem, on the other end, this cable is branched to have one DB9 female connector and one power jack as shown in the Figure 2.5.



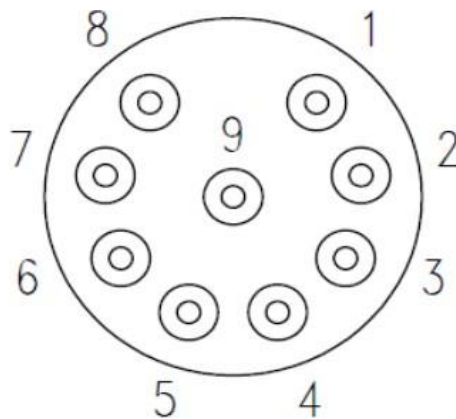
**Figure 2. 5 Acoustic modem connector**

The raw cable has 8 wires and a shield. Only 5 wires (black, white, red, green and grey) and the shield are used by the modem to get power and communicate with other devices using RS-232 protocol. Three wires and the shield are connected to power and ground, two wires are used to transmit and receive for RS-232 interface. The Table 2.1 shows the pin connection on the DB9 connector for each wire.

Signal	Wire Color	Pin Number
Power(12VDC ~ 16.5VDC)	Black and White	1 and 2
Ground	Red and Sheild	3 and 9
Modem transmit data	Green	4
Modem receive data	Grey	6

**Table 2. 1 Connector pin details**

If the cable is detached from the modem, the connector pin is numbered as shown in Figure 2.6



**Figure 2. 6 Acoustic modem connector pin**

The pinout shown in Figure 2.6 is especially important when interfacing with the water proof housing. Interfacing these cables with the Raspberry pi needs a USB to Serial converter. Once the cable is plugged into the Raspberry pi, one should open any serial emulator like minicom and use the following settings as shown in Table 2.2 to initialize communication.

Baud rate	9600 bps
Data bits	8
Stop bit	1
Parity	None
Flow control	None

**Table 2. 2 Acoustic modem serial settings**

### Acoustic modem interface with Raspberry pi

- Find a DC power supply that can provide 12VDC ~ 16VDC(max 60 watts), and connect the power to the supplied power jack. Make sure that the power is connected to the terminal with “+” sign and ground is connected to the terminal with “-” sign. Check power

polarity on the power plug side (inside is power and outside is ground), then you plug it into the power jack on the waterproof cable.

- Connect the DB9 connector of the waterproof cable to a computer. Setup a terminal tool for the corresponding serial port.
- Plug the other end of the waterproof cable into modem. (Make sure it is fully seated). The following message should print out on the serial port emulator.

*AQ\_MODEM\_AM12.*

*Modem will start in 10 seconds.*

*Press S for system check before regular boot.*

*Press F to go to firmware utility.*

10 seconds later, more messages will appear on the terminal emulator

*MID:3*

*TXPWR: -50*

*RXGAIN: 0*

*RTC:2013-12-12T13:01:54*

*UIMODE:16(Command)*

*CMDTERM:CRLF(I),CRLF(O)*

*ESC:^^^A*

*LPMODE:0*

*TSTDB:120*

*BIV:AM96121013(C7), AM96121013(C5)*

- Send the following command to test the modem ,each command should end with carriage return and line feed.

*\$HHTXA,0,1,0, hello world!*

- The default transmitting power is set to -40dB of full power. The transmission may be audible at that level. To change the power level, use the following command.

*\$HHCRW, TXPWR, -20*

- The modem has the capability of transmitting both text messages as well as waveforms depending upon the commands. It's given the modems internal processor calculates some of the signal parameters, such as SNR, Input SNR, Effective SNR etc.

## RF modem

The RF modem shown in Figure 2.7 (Microhard n920-ENC specifications shown in Table 2.3) used is communicate with the external world to acquire the control parameters for the acoustic modem. The RF modem is essential component of the node next to acoustic modem. The RF modems from different nodes form a point to multi point communication network. In this method of networking there is be one master and multiple slave.



Figure 2.7 RF modem

Frequency	902 – 928 MHz
Spreading Method	Frequency Hopping /DTS
Output Power	100mW to 1W(20-30 dBm)
Sensitivity	-166dBm – slow, -108dBm – fast, -100 dBm-Turbo
Serial Baud Rate:	Up to 230.4 kbps – Slow/Fast Up to 3.2 Mbps - Turbo
Link Rate	19.2 kbps to 1.3824 Mbps
Core Voltage	3.3VDC Nominal
Error Detection	32 bits of CRC, ARQ

Table 2. 3 RF Modem Specifications

### Interfacing RF Modem with Raspberry pi

- The procedure uses two RF Modems. Connect the USB interface(this does not power up the RF Modem it is just a data line) to Raspberry Pi.
- Open a serial connection software (minicom, putty.etc) session for each modem connection, and configure the serial emulator settings as shown in Table 2.2. Then open the ‘connection’.
- Provide power to the RF modem using the DC power supply (8-30V DC) which comes with the modem, while pressing the config button which is on the front side of the modem

as shown in Figure 2.7. And as soon as the adapter is plugged in release the button and a blue light appears in front of the modem. This means that the modem is in command mode. Do the same for the other modem also.

- Now type AT&F6, AT&WA command consecutively for one of the modem, which makes it master node. Enter the commands, the AT&F7, AT&WA consecutively for the second modem to make it slave. After few seconds the RSSI leds will glow on both modems indicating that they are connected (now the modem is in data mode).
- Letters entered in one session window appear on the other window. Sometimes the RSSI leds will glow on one modem. At that time, only one way transmission is possible. Unplug and plug the modems (no need of any configurations to be done this time). Then both modem's RSSI leds glow. Indicating two-way communication.
- If we want to change from data mode to command mode enter the escape string which is '+++'

### Configuring the RF Modem's registers

Before proceeding to this section make sure that the RF modem is in command mode as detailed in previous paragraph. Here is the method to configure the registers after connecting to the modem via any serial emulator.

The general command structure is:

Query format: **ATS<S register #>? [Enter]**

Change format: **ATS<S register #>=<value> [Enter]**

#### S0 Default on state

This register determines in which mode the modem will be upon power-up. If selected to power-up in Command Mode, the modem will be offline from the wireless network, and ready to be configured upon power-up. The typical mode of operation is for the modem to power-up in Data mode: ready to participate in data transfer over the wireless network.

Values	Action
0	Command Mode
1	Data Mode

**Table 2. 4 Default on state settings**

#### S102 Serial Baud Rate

The serial baud rate is the rate at which the modem is to communicate with the attached local asynchronous device. The available baud rates are shown in Table 2.5

Values	Action
1	115200
5	19200
7	9600
9	4800
12	1200

**Table 2. 5 Serial Baud Rate settings**

### **S103 Wireless Link Rate**

This register determines the rate at which RF communications will occur over a given network. All modems within a particular network must be configured with the same wireless link rate. Faster link rates result in greater throughput, however, for each 'step' increase in link rate, there is an approximately 1dB reduction in sensitivity.

Values	Action
0	19200
1	115200
2	172800
3	230400
8	1.2 Mbps

**Table 2. 6 Wireless Link Rate**

### **S108 Output Power**

This setting establishes the transmit power level which will be presented to the antenna connector at the rear of the modem. Unless required S108 should be set not for maximum, but rather for the minimum value required to maintain an adequate system fade margin.

Values	Action(milli watt)
20	100
22	160
24	250
26	400
28	630
30	1000

**Table 2. 7 Output Power settings**

## S123 Average RSSI

This register displays (it is not a 'setting') the average signal strength received over the previous 4 hop intervals. The value in this register is also reflected in status lines RSSI1, 2, and 3, which connect to the modem's front panel RSSI LEDs. There are many other registers used by the RF modem. TO learn about the remaining registers and their control settings please check the manual.

## Antenna for the RF modem

There are two types of antennas which are used for each node depending upon the experimental requirements. The two antennas are.

- L-COM HyperLink Wireless 900 MHz 8 dBi Professional High-Performance Omni Antenna Model HG908U-PRO
- AIR802 902-928 MHz 2.5 dBi Dipole (Rubber Duck) Antenna with RP-SMA Connector.

The omnidirectional antenna is used because we need the radiation pattern to be symmetrical. This provides equal signal gain in every direction. The Rubber Ducky antenna is useful when we experiment in the lab. Using these antennas and RF modem Li Wei, Wensheng Sun, Chaofeng Wang, Krishna Chaitanya Poduru conducted a test to find the range vs RSSI value for the above-mentioned combination of antennas and RF modem.

- **RF Modem Used:** microhard n920-ENC
- **Antenna make and model:** L-COM HyperLink Wireless 900 MHz 8 dBi Professional High-Performance Omni Antenna Model **HG908U-PRO**, **AIR802** 902-928 MHz 2.5 dBi Dipole (Rubber Duck) Antenna with RP-SMA Connector
- **Experiment Process Description:** The RSSI value on RF modem is recorded as the index of link quality. On the first RF modem which acted as RX we checked register S123 value where we can find RSSI value on RF modem, whose value ranges from -55 to -110 dBm. On the Second RF modem which worked as TX we set the output power of it to 1000 mW by changing the value of register on RF modem is S108 to 30. Then we went to various locations as shown in Figures 2.8 and Figure 2.9 keeping one end as TX ,other end as RX and transmitted few characters to obtain RSSI values from RX

**Results(courtesy of Li Wie):** HG908U denotes the big 8dBi antenna, and AIR802 denotes the small 2.5dBi antenna.



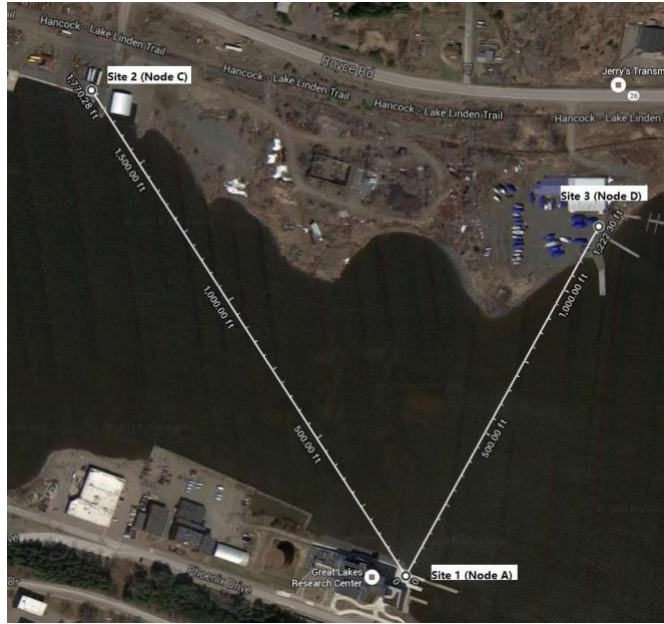


Figure 2. 8 Keweenaw waterway test site

Between Site 1 and 2	RSSI at Site 1(dBm)	RSSI at Site 2(dBm)
HG908U	-51	-51
AIR802	-91	-94

Between Site 1 and 3	RSSI at Site 1(dBm)	RSSI at Site 3(dBm)
HG908U	-52	-55
AIR802	-79	-90

Table 2. 8 RF Modem RSSI Values at Keweenaw Waterway Test site

### Portage Lake Test

Test the RF link with both types of antenna for cross the whole lake and cross half of the lake. The test locations are shown in Figure 2.9 and 2.8:

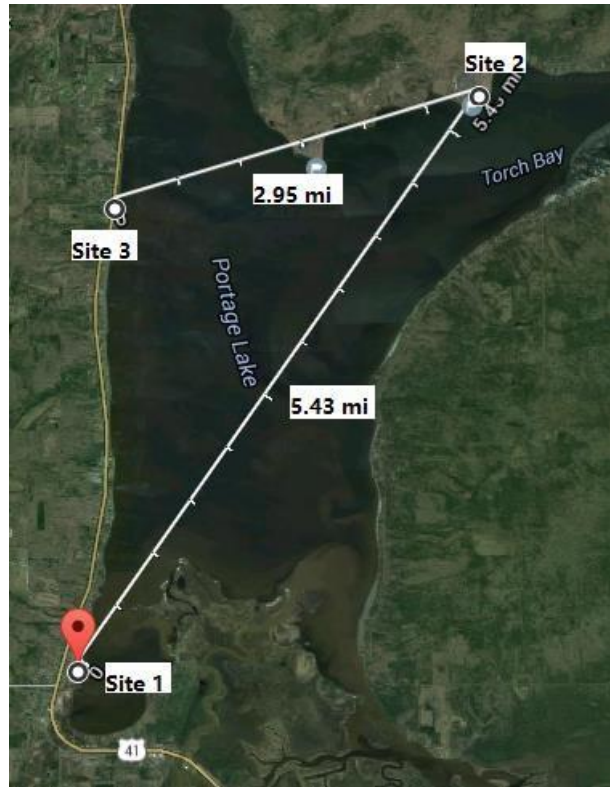


Figure 2. 9 Portage lake test site

Between Site 1 and 2	RSSI at Site 1(dBm)	RSSI at Site 2(dBm)
HG908U	-93	-100
AIR802	-104	-105

Between Site 2 and 3	RSSI at Site 2(dBm)	RSSI at Site 3(dBm)
HG908U	-93	-87
AIR802	-105	-103

Table 2. 9 RF Modem RSSI Values at Portage Lake Test site

**Conclusion:**

Both types of antenna could be used in future Keweenaw Waterway test, the HG908U 8dBi antenna is recommended for field test in Portage lake or other field test within 5 miles region.

The results are for sunny days and the antenna is not shaking. Link quality could be worse in unpleasant weather or the antenna is shaking.

2.2.3. Battery subsystem

The battery subsystem shown in Figure 2.10 consists of a LiNiMnCo 26650 Battery Pack, which is rated at 14.8 V and 8Ah. This battery module consists of 8 pieces of LiNiMnCo 4000 mAh cylindrical 26650 cell in 4S2P (2Rx4C) configuration which is two sets of 4 batteries each in series

which are in turn in parallel. This group of batteries are enclosed in Aluminum Enclosure: 125 x68x84 (mm)-(Bx-AC-120062-68).



**Figure 2. 10 Battery pack**

#### 2.2.4. Voltage regulation subsystem

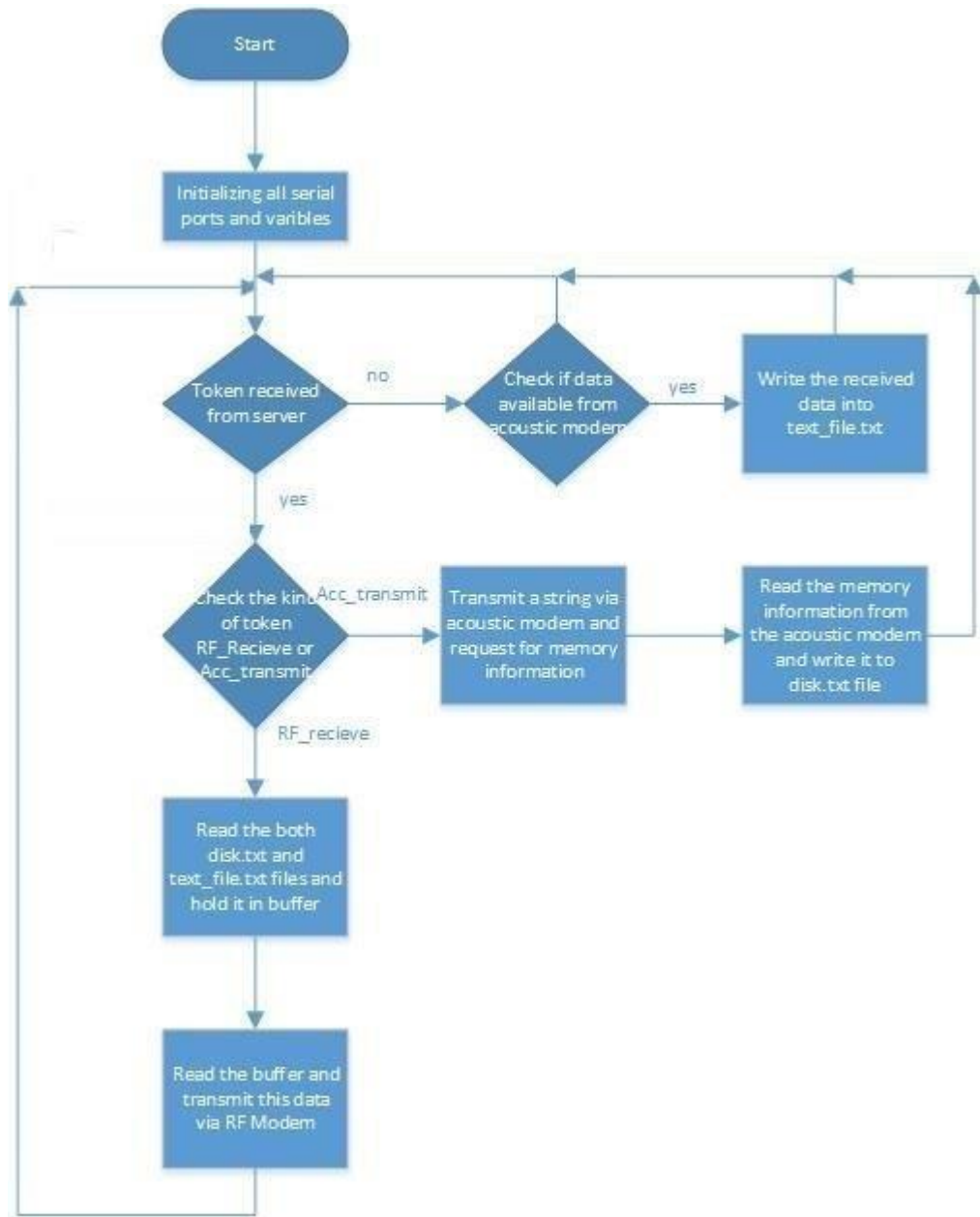
The voltage regulation subsystem purpose is to give the respective components their designated rated voltages. This system consists of two DC-DC buck converters, which are rated at 8-22V for input and 1-15V as output and 3 amperes of current with an adjustable screw which is used to adjust the output voltage. Some of the mechanical details of it are 1.3 x 1 x 0.8 inches dimensions and 1.6 ounces weight. Most important information with these types of converters are The input voltage must be set at 3 volts greater than the desired output voltage. One of the two DC-DC converters is set to output a 5 V dc supply for Raspberry pi. And other is set at 12V dc supply for the RF Modem.



**Figure 2. 11 DC-DC converters**

### 2.3. Single Node System operation

The primary operation of each node is to get the acoustic channel data of the water column by performing underwater communication. So, the rest of the node architecture is primarily based to transmit this acoustic channel data to control center. The computing platform for each single node is Raspberry pi which is based on the ARM Cortex V7 architecture. The operating system is a Linux distribution named Raspbian. All the programs are written in the Python programming language. The control algorithm is shown below in Figure 3.0. Both the RF modem and acoustic modem are interfaced to the USB serial hub which in turn is connected to the Raspberry pi USB port. In the first segment of the code all the initializations are made the modem transmit power is set to a predefined power level, From then the code runs in an infinite loop structure. It checks for any messages received from server. If it didn't receive any messages, then the Raspberry pi checks to see if there is any data available in the serial port input buffer. If it finds any data in the buffer then it reads the data and puts it in a file. If not, then it goes back to the system state where it checks for the received messages from server. If it receives any request messages from server then it checks for kind of requests which are whether it is request-for-acoustic transmission message or request-for-data message. If it's a request-for-acoustic transmission message, then the Raspberry pi writes two strings to the input buffer of the acoustic modem. The first string is a transmit message like hello world and the second string is a request command for memory information of the acoustic modem. Later in this line the Raspberry pi stores the memory information in a second file. If the received message from the server is request-for-data message, then the Raspberry pi reads both the files and then sends the information to RF modem which then transmits the information to the base station.



**Figure 2. 12 single node system operation**

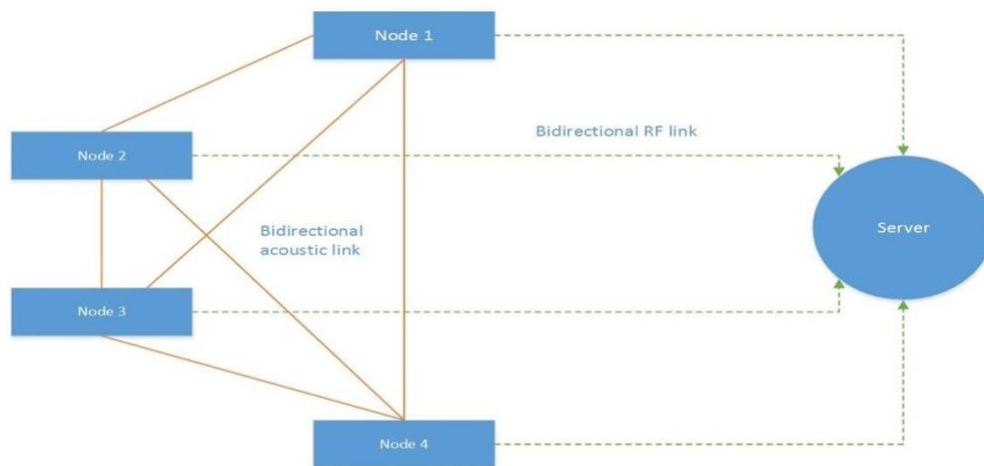
### 3. TESTBED ARCHITECTURE FOR MONITORING THE SPATIAL-TEMPORAL DYNAMICS OF UNDERWATER ACOUSTIC CHANNELS

This chapter describes the network structure of the system which includes 4-remote nodes and a server. This chapter also deals with the server layer design which is the most crucial system design aspect because majority of the system control decisions are made by this subsystem. This chapter includes networking strategies, error handling measures, MySQL database interface. This entire level corresponds to the backend.

#### 3.1. Overview of the testbed architecture

There are many motivations for introducing this subsystem. One of the primary reasons is to get the acoustic channel data from all the nodes in real time. Another important motivation is to have a common clock for all the nodes which is fulfilled by the server. Other reasons are the need for a platform to provide secure shell(ssh) login to all remote nodes in order to get complete control of the node. The system is based on the dual-layer message system which is more efficient, easy to implement, and meets the demands of our application. The complete layout of the network is shown in Figure 3.1 below.

The four nodes are connected in a mesh topology fashion using acoustic links. The server forms a star topology with all remote nodes using RF links. The reason for the mesh topology between the nodes is choice by application because we want every acoustic node to broadcast the information so that we will be having all links data at once. The reason for the star topology is due to the limitation of the RF Modem capability. Both the RF and Acoustic links are bidirectional. Each node consists of a Raspberry pi, an RF modem, an acoustic modem and supporting hardware. The server consists of a powerful desktop computer and an RF modem for communication with the nodes.



**Figure 3. 1 High level view of server and node connection**

### 3.2. Server level architecture and algorithm

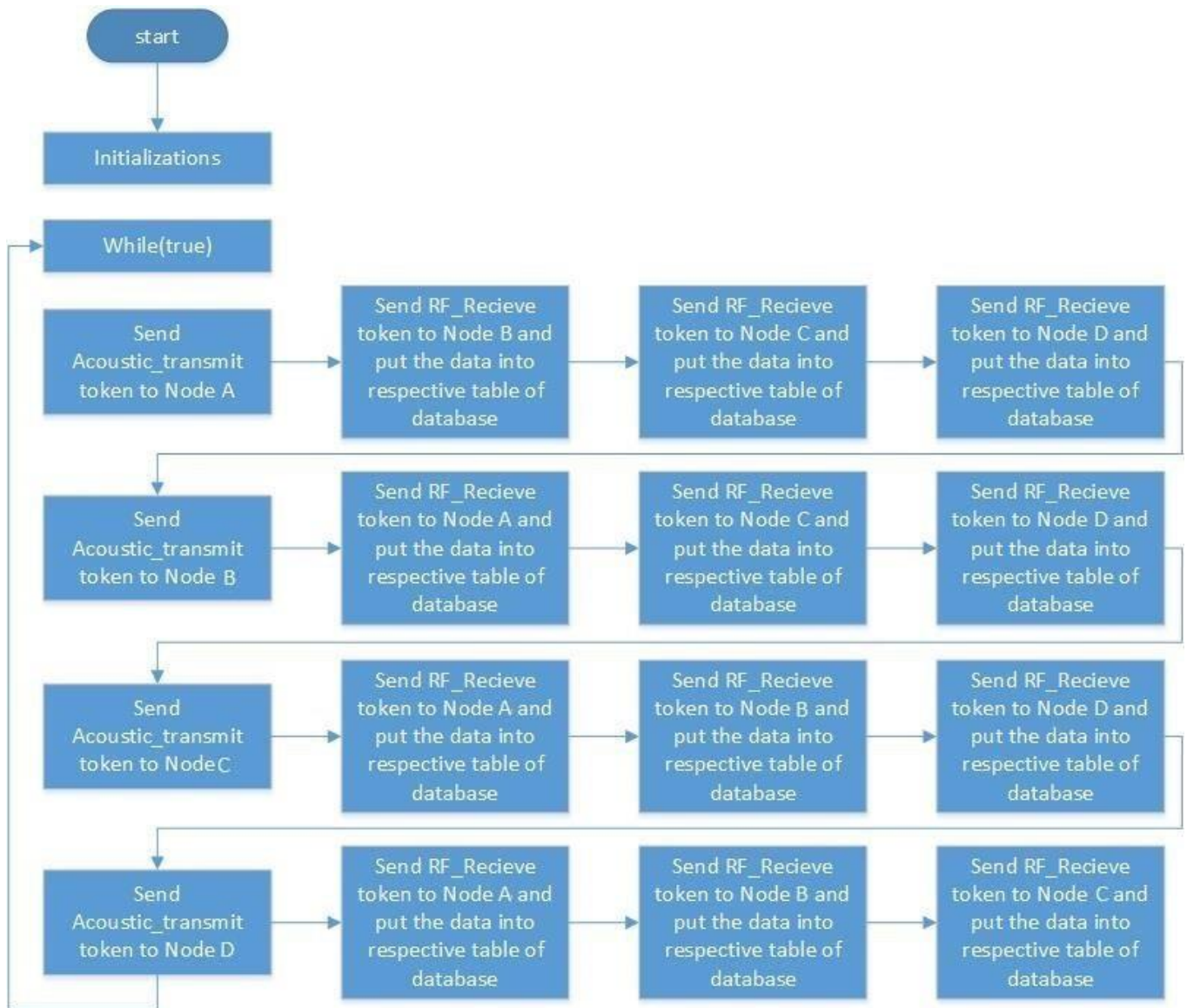
The server is designed to meet the following demands.

- It acts as a central clock for all nodes.
- It generates the messages based on a TDMA scheme.
- It downloads the acoustic channel data from all the remote nodes.
- It extracts all the required information, segregates the required data, and puts it the correct columns of the MySQL database.
- It provides exception-handling mechanisms to improve the stability of the system.

The following paragraphs describes how the above demands are met by the design. The server is interfaced to a RF Modem via USB to Serial link. The server algorithm, which does all the work, is written in MATLAB. The dual layer message based system means that there are two kinds of messages Request-for-data message and Request-for-acoustic transmission message. Request-for-acoustic transmission is a message type sent from the server to the node to receive the data acquired by the node, to the modem via RF link. Request-for-acoustic transmission message is a message, which is sent from the server to the node to make the node broadcast a message via acoustic modem. The RF modem is configured in point to multipoint (P2M) settings. In this setting, all the nodes receive the message but depending upon the type of message only one node receives whatever the server sends and takes action. The rest of the nodes reject the message. The request-for-data message is required because when any remote node broadcasts a message via acoustic modem, all the remote nodes receive it Every node has data, which is needs to be communicated to the server. These kinds of messages are introduced to avoid collision and regulate the traffic.

The entire experiment setup is based on a round robin transmission. Therefore, the server sends the request-for-acoustic transmission message by RF modem to every node consecutively. The request-for-data message are also issued in same manner to receive data. The received data from any node is then extracted and segregated into respective columns based on the string compare technique i.e. it looks for some key strings in incoming data and extract the values beside it. when all the row details are filled, it is written to the database. Figure 3.2 below shows the flow chart of the processing algorithm. Initially the server configures the serial port and connects to the database. The server then sends request-for-acoustic transmission message to Node A, which makes it broadcast a pre-loaded string, or waveform. After a certain amount delay, all the other nodes receive this data and are ready to transmit this data to the base station (i.e. server). Next the server issues a request-for-acoustic transmission to Node B to get the data related to the A—B channel. After receiving the complete packet, it extracts the respective fields and puts them in the database. In a similar fashion the server sends request-for-data message to the rest of the nodes, C and D for data related to channels A—C and A—D. The server sends request-for-acoustic transmission message to B and the procedure goes on. The server also checks for any missing data and put zeros in place of the missing fields. This procedure is continued as shown in Figure 3.2.





**Figure 3. 2 Server level architecture and algorithm**

### 3.3. Stability and error handling mechanisms

System stability is a major concern. Whenever there is some error or malfunction at any node must not lead to crash the program running on the server. So, to deal with these errors, exception handling mechanisms are used to increase the stability of the program. Whenever the RF link quality between the server and the remote node is not poor some of the data the server is expecting is lost and it crashes the program. The way the server segregates the data is based on the string comparison mechanisms. The server always tries to look for some key strings in the incoming data. If any of the data is missing, then the server, using exception handling mechanisms, will put zeros in place of the missing data.



### 3.4. Interfacing MySQL database with MATLAB

The primary need for the database is to store the data for future reference. It acts as a base for the GUI, from which it gets the data to visualize it for the user. The same function can be achieved by storing the data in MS Excel format. For the current application, the MySQL database is used and is installed in the server. It runs concurrently with MATLAB script. DB Explorer is the MATLAB toolbox, used to interface the MATLAB script with the MYSQL Database. The toolbox uses JAVA Database Connectivity(JDBC) drivers to connect with the MySQL. Here are the steps one need to take for interfacing both.

1. Download the JDBC connector for MySQL database and save it in local system.
2. In the command window of the MATLAB, type “javaaddpath directory of the connector” which adds the JDBC connector to MATLAB.
3. This enables communications with the database. Type “dexlore” in the command window. This produces the GUI version of the toolbox. Enter the address, port, username, password and database fields to make a connection.

The current database has 12 tables. Each corresponding to the respective link i.e. A-B, A-C, A-D, B-A etc. Each table have 18 columns, which includes all the data corresponding to the acoustic channel. To insert the data into the database use “datainsert” command.

## 4. GUI DEVELOPMENT

This chapter deals with the construction of different sub layers of the GUI architecture. The following chapter includes interfacing the MySQL database with JAVA, JSON object creation of data using JAVA, using JAVA script, p5.js, HTML and other libraries for creation of the web GUI, hosting the website using apache server and complete client architecture briefly.

### 4.1. Introduction and architecture

This subsystem is used to visualize data, which is very important to observe patterns or gain intuition about the properties of underwater acoustic channels. In addition, it provides the distant user the ability to look up on the data in real time. The GUI will also be used to control the system in real-time, in the future scope. There are two different GUI systems. One, runs at server level. The second is a web based GUI which can be accessed from a web browser. There are several reasons for having two GUI'S at various levels. One is to provide security i.e. one can only control the system from the server level GUI. It can be used for both viewing and controlling many parameters. Whereas one can only view the graphical data from web based GUI. Another reason is the server level GUI subsystem yields JSON objects that contains data used by the web based GUI as a source of information. The Figure 4.1 depicts the entire GUI subsystem starting from the MySQL database the data flows both to MATLAB GUI and JAVA GUI. The JAVA based GUI provides a base platform for controlling the device and converts the data from database to JSON Objects. Then JAVA Script along with other libraries is used to create the web based GUI later hosted in the Apache <sup>[7]</sup> server. The client can be any web browser accessing the server.

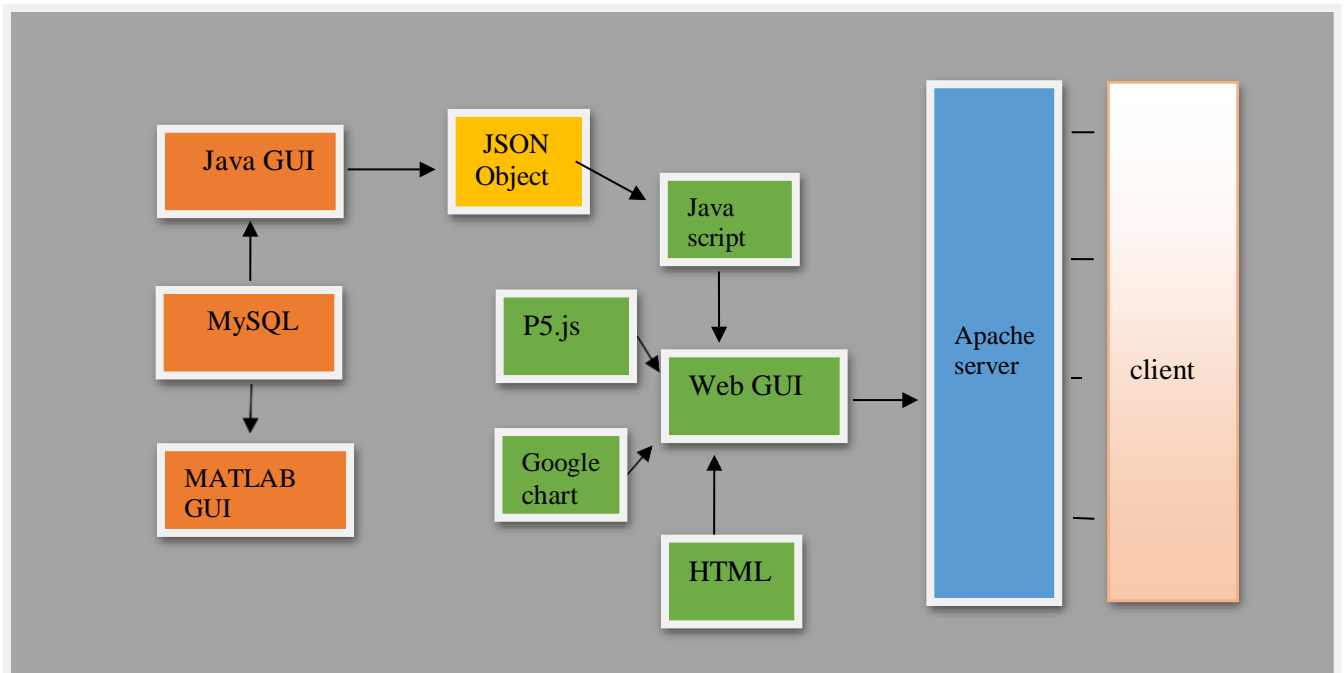


Figure 4. 1 GUI srchitecture

## 4.2. GUI at server level

There are again two GUI's at server level. One uses MATLAB and the other uses JAVA. The one using MATLAB, extract the data from MySQL database and plots it. The GUI script is runs alongside the server algorithms. Using this type of GUI's leaves many possibilities for the future eg. Making prediction analysis using machine learning and projecting in the same GUI etc.

The second GUI is made using JAVA. There are two primary functions for this GUI. One is to create the JSON objects from the database. These data objects are later used by the web based GUI for graphing the JSON object which is updated every second to give real-time flexibility. The second function is to show graphs of data and control the system (future scope). JSON objects have many important uses. In the future one can build a web API which uses the same objects. And JSON formatted data helps to reduce the development time for web GUI. The JAVA uses the JDBC drivers to communicate with MySQL database.

## 4.3. Web GUI

The web based GUI is created using javascript. It uses two libraries one is P5.js and the second is google charts api. p5.js is used to create the buttons and other layouts. Google charts is used for creating the graph. Google charts api takes the JSON objects created by the server level GUI and projects them on the webpage as soon as one clicks the corresponding button. The GUI is shown in Figure 4.2. The website is hosted using apache server which is hosted in server.

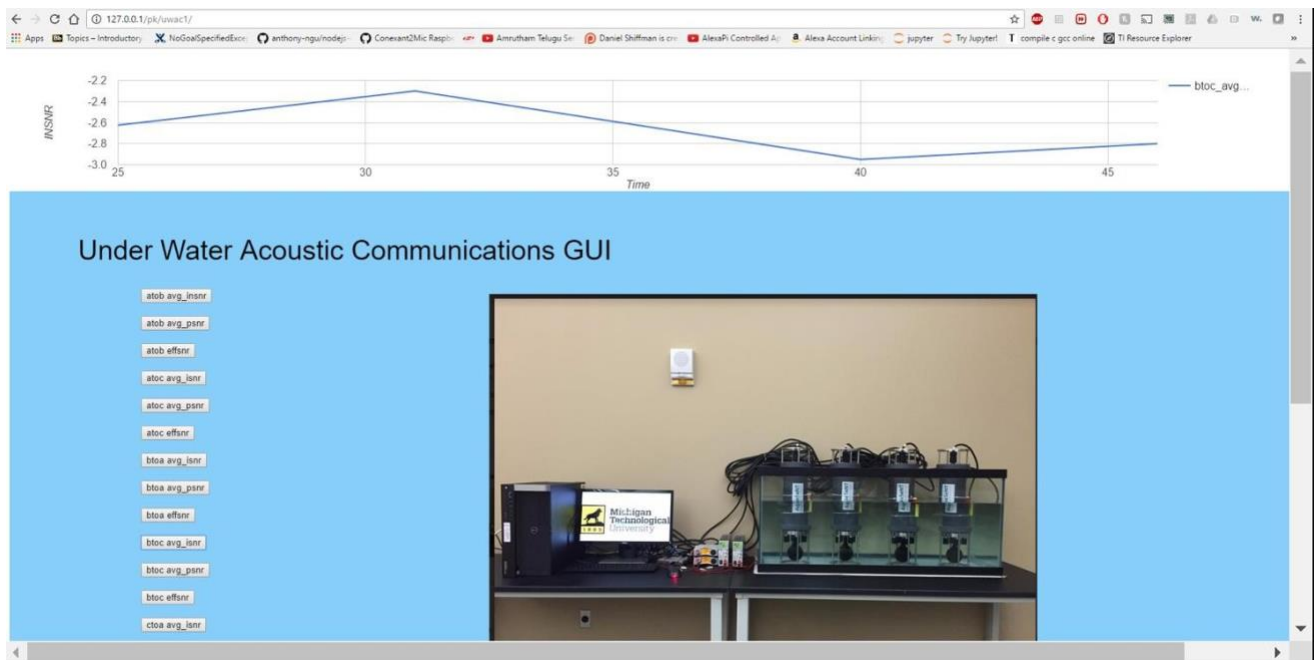
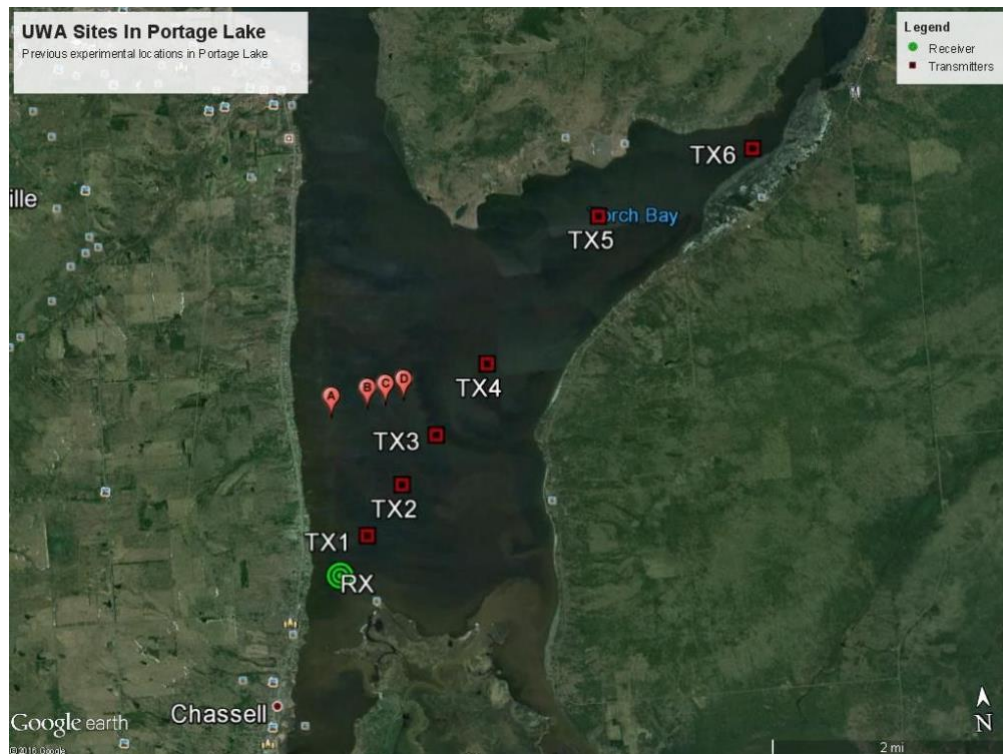


Figure 4. 2 Web GUI

## 5. A FIELD EXPERIMENT IN THE PORTAGE LAKE

### 5.1. Experiment overview

This experiment was conducted on September 30<sup>th</sup> of 2016 on Portage Lake [25]. There are two major tasks in this experiment, compare the results of the current experiment with similar experiment which is conducted on June of 2014 (the coordinates of the experiment are found on below Figure 5.2) and second task is to compare the current experiment with the one which is conducted on March of 2016 (coordinates of the experiment is found on Figure 5.3). The entire experimented locations are shown in Figure 5.1. Locations A, B, C, D are related to task 1 and rest of the other are related to task 2



**Figure 5. 1 Experiment overview**

Distance [m]	0	500	750	1000
Latitude	47°3'45.36"	47°3'47.82"	47°3'48.94"	47°3'50.34"
Longitude	88°30'42.36"	88°30'18.78"	88°30'6.97"	88°29'55.08"
Water depth [m]	11.5	10.2	15.2	14.1
Modem depth [m]	5	5	5	5

**Figure 5. 2 (June 2014)**

	RX	TX1	TX2	TX3	TX4	TX5	TX6
Coordinates N <sup>2</sup>	47°02.596'	47°02.882'	47°03.232'	47°03.575'	47°04.065'	47°05.085'	47°05.493'
Coordinates W	88°30.735'	88°30.401'	88°29.989'	88°29.583'	88°28.976'	88°27.648'	88°25.926'
Distance [m]	N/A	684.08	1524.25	2344.88	3501.12	6075.41	8099.81
Average Received SNR [dB]	N/A	26.0575	27.9778	27.9972	27.4050	27.7850	23.6444
Average Pilot SNR [dB]	N/A	23.04	23.33	24.0056	23.9650	22.41	23.6972

**Figure 5. 3 (March 2016)**

## 5.2. Experimental procedure

Task one uses two nodes. The Rx node is a complete node (i.e. with Acoustic modem (modem 34), RF Modem, Raspberry pi etc.) and Tx node (modem 35) is just the Acoustic Modem.

1. First the Rx node was placed at location A in the Figure 5.1. Then Tx node was taken to location B and a waveform was sent from Tx node to Rx node. Then the acoustic information which was stored in B is extracted via RF Link by sending its request-for-data message manually.
2. Now the transmitter node is taken to location B and the same procedure has been done. Using the ranging function to detect Rx node, the same experiment is continued with location C and furthermore till approximately 6000 meters.

For task 2 we had done the same experiment which was carried out in March 2016. The following are the set of power levels used -50, -40, -30, -20, -10, 0 dB. Below are some facts and key terms which were used in this experimentation

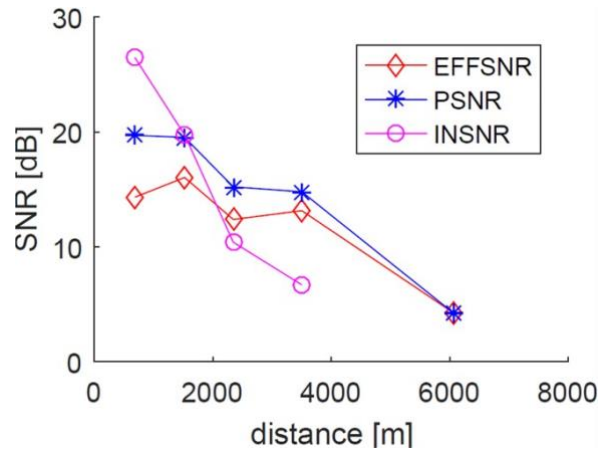
1. 10 Type 1 transmissions (text messages) with each power levels. (estimated time is: 3 minutes and 20 seconds)
2. 15 Type 2 waveforms at each power level (estimated time: 16 minutes)
3. 40 Type 3 wave forms at -10 dB at two locations (estimated time :30 minutes)

And task 2 was carried out this way

1. First went to the Rx location in above Figure 5.3, deployed the Rx node associated with modem 47, adjusted to 4.5 meters
2. Went to location TX1 in Figure 5.3, performed two types of transmission. Got the data from Rx node via RF link.
3. Went to location TX2 in Figure 5.3, perform three types of transmission, then deployed node associated with 41.
4. Went to location TX3 in Figure 5.3, performed two types of transmission.

5. Went to location TX4 in Figure 5.3, performed three types of transmission, then deployed node associated with 34.
6. Went to TX5 and TX6 and perform two types of transmission.

### 5.3. Results



**Figure 5. 4 Results of portage lake experiment [courtesy of Wensheng Sun]**

The above Figure 5.4 is the graph in comparison with distance on x axis and various SNR's at a constant depth of Rx and Tx Nodes. If one observes the Pilot SNR(PSNR) is constantly in higher magnitudes in comparison to Effective SNR(EFFSNR). A sharp drop of Input SNR(INSNR) is seen in comparison to EFFSNR and PSNR.



## 6. KEWEENAW SPATIAL-TEMPORAL EXPERIMENT

### 6.1. Experiment overview and setup

The current experiment is conducted in Portage Lake near Michigan Tech between 22-24 April 2016. Four nodes with a Raspberry pi and Acoustic modem on each node were deployed at these places: GLRC dock (A), Facility Management (B), Julio contracting company (C) and H&Y marina (D). Six links are formed “ab”, “ad”, “ac”, “bd”, “bc”, “cd” which includes two Non-line of sight (NLOS) links, namely “ab” and “cd”. Node A and B are powered up using a power adapter during the experiment, node C and D are power by batteries. Node A and B will sent out two consecutive waveforms within one 12 minutes cycle , the time interval between two waveforms is one minute. However, only one waveform was sent by C and D in each cycle.

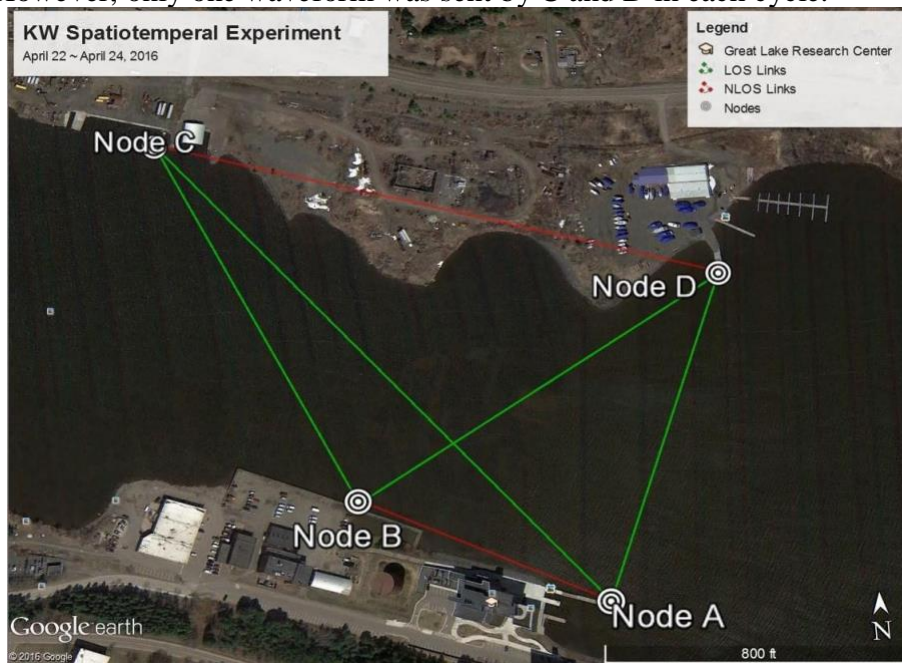
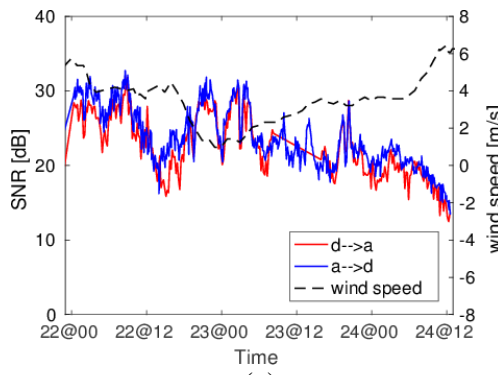


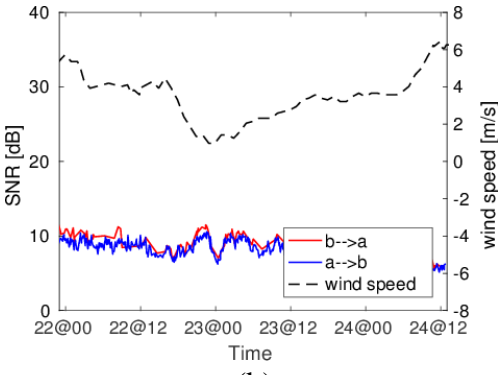
Figure 6. 1 Experiment setup

### 6.2. Results

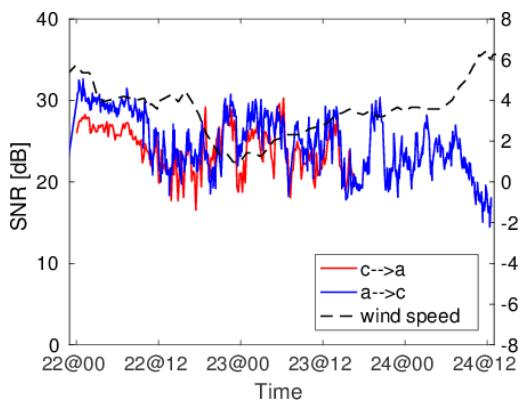
The below Figure 6.2 shows the relation between PSNR's with respect to time for different links. The covariance matrix between all links are shown in figure below, besides the auto-covariances, the reciprocal links show very high correlation (around 0.8). The histogram of the correlation coefficients without auto-covariances are shown in figure below the median is above 0.5.



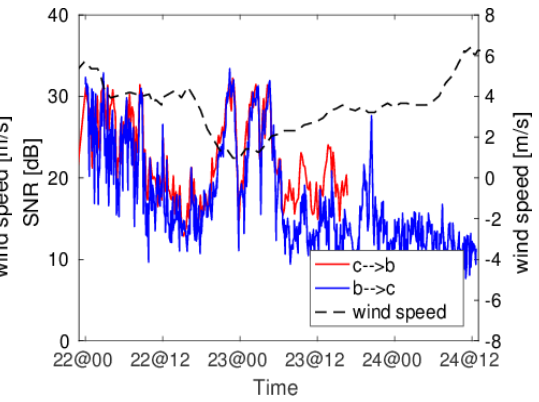
(a)



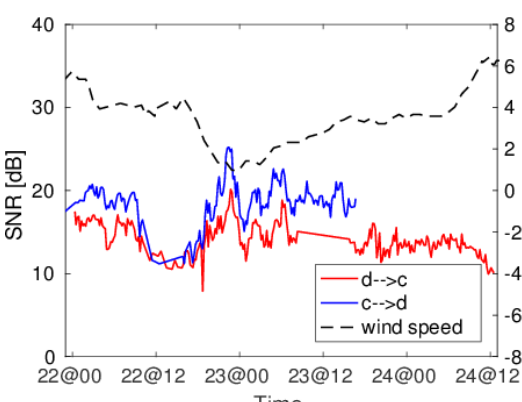
(b)



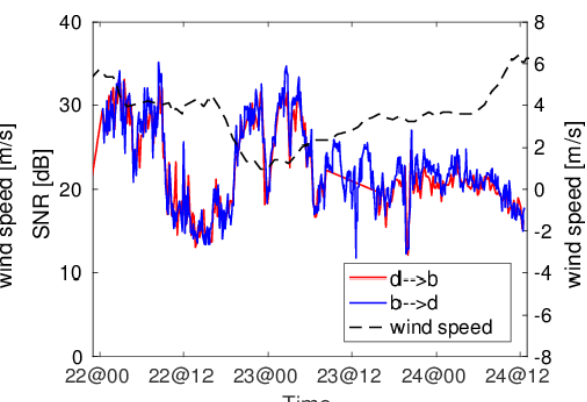
(c)



(d)

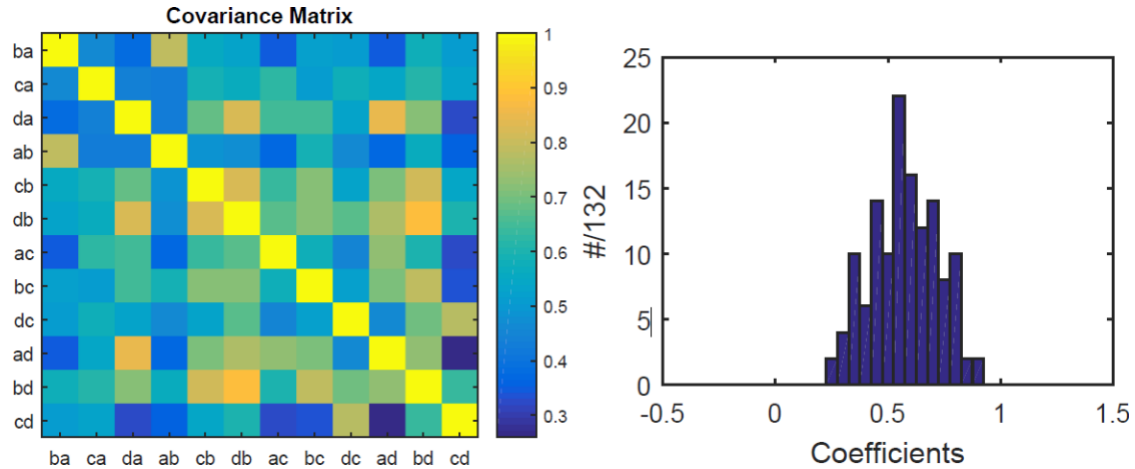


(e)



(f)





**Figure 6. 2 Keweenaw spatial-temporal experiment [courtesy of Wensheng Sun]**

### 6.3. Problems faced and solutions

- Node C runs out of memory after few transmissions (since node C receives higher number of messages) thus the newly received message overwrites the old ones. And a whole subfolder is erased. This problem can be solved with existing node design since it transmits the data continuously to the base station. So, we can clear onboard acoustic modem memory using Raspberry pi as soon as data is transmitted.
- The battery needs to be changed frequently. Sometimes due to outside temperature the battery drained too fast. So, By the time we reached the site to change the battery, the node stopped operating. If a minor change is made to the existing design by adding coulomb meter, charge of battery information could be sent via RF link in the same packet of information. We can monitor the battery levels and replace them.

## 7. CONCLUSIONS AND FUTURE WORK

There are several places at system every level where improvements and new features can be added, for example

1. If one includes a coulomb meter at the node and transmit its status to the base station so one can know how much power is left in the battery and plan accordingly.
2. Include an algorithm to support multi-hop communication to increase the communication range of experimentation.
3. Using a custom Linux distribution helps in increasing the efficiency of system.
4. Make the existing code interrupt based rather than polling mechanism which reduces a lot of computation bandwidth and power.

Server level

1. Improve the algorithm to support multi-hop communication in case of node failures.
2. Improve error handling mechanisms to request missing data.

GUI level

1. Remove the redundant GUI's depending upon the usage. This can reduce the complexity at this level.
2. Develop JAVA GUI by adding additional functionality and remove the dependency on web GUI by making web GUI to query the database directly.

## 8. REFERENCES

- [1] L. Wei, P. Zheng, H. Zhou, J-H. Cui, S. Zhou, Z. Shi and J. O’Donnell. “Long Island Sound Testbed and Experiments “. Department of Computer Science and Engineering, University of Connecticut, Storrs.
- [2] L. Wei, Z. Wang, J. Liu, Z. Peng, and J-H. Cui. “ Power Efficient Deployment Planning for Wireless Oceanographic Systems”. IEEE SYSTEMS JOURNAL.2016. IEEE.
- [3] E. Demirors, J. Shi, R. Guida, and T. Melodia. “SEANet G2: Toward a High-Data-Rate Software-Defined Underwater Acoustic Networking Platform”. WUWNET ’16 October 24-26, 2016, Shanghai, China.
- [4] YMODEM, <https://en.wikipedia.org/wiki/YMODEM>
- [5] Gumstix, 2004-2012 Gumstix inc, <https://www.gumstix.com/>, 2013.
- [6] Raspberry pi , [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [7] Apache server, <https://httpd.apache.org/>
- [8] D. B. Kilfoyle and A. B. Baggeroer, “The state of the art in underwater acoustic telemetry,” IEEE J. Oceanic Eng. 25, 4–27 (2000).
- [9] T. C. Yang , “ Properties of underwater acoustic communication channels in shallow water” , Naval Research Laboratory, 4555 Overlook Avenue, S.W., Washington, DC 20375
- [10] Aquasent , <http://www.aquasent.com/acoustic-modems/>
- [11] Y. Sun and T. Melodia, “The Internet Underwater: An IP-compatible Protocol Stack for Commercial Undersea Modems,” in Proc. of ACM Intl. Conf. on UnderWater Networks and Systems (WUWNet), Kaohsiung, Taiwan, November 2013.
- [12] Raspbian, <https://www.raspberrypi.org/downloads/raspbian/>
- [13] JTAG, <https://en.wikipedia.org/wiki/JTAG>
- [14] Yocto project, <https://www.yoctoproject.org/>
- [15] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: Research challenges,” *Ad Hoc Networks*, pp. 1–23, Jan. 2005.
- [16] A. B. Baggeroer, “An overview of acoustic communications from 2000-2012,” in *Proc. of the Workshop on Underwater Communications: Channel Modelling & Validation*, Italy, Sept. 2012.
- [17] M. Chitre, S. Shahabudeen, and M. Stojanovic, “Underwater acoustic communications and networking: Recent advances and future challenges,” *Marine Technology Society Journal*, vol. 42, no. 1
- [18] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, “The challenges of building mobile underwater wireless networks for aquatic applications,” *IEEE Network, Special Issue on Wireless Sensor Networking*, vol. 20, no. 3, pp. 12–18, May 2006.
- [19] [https://en.wikipedia.org/wiki/Underwater\\_acoustics](https://en.wikipedia.org/wiki/Underwater_acoustics)

- [20] Zhaohui wang, Chaofeng Wang, Wensheng Sun, "Adaptive transmission scheduling in time-varying underwater acoustic channels" , OCEANS'15 MTS/IEEE Washington, 19-22 Oct. 2015, Washington, DC, USA.
- [21] ORTUN, [https://www.isi.edu/ilense/ortun/#Related\\_Links](https://www.isi.edu/ilense/ortun/#Related_Links)
- [22] WHOI micromodem, <https://acomms.whoi.edu/>
- [23] SNUSE modem, <https://www.isi.edu/~johnh/PAPERS/Syed08c.pdf>
- [24] W. Sun, and Z.-H. Wang, "Modeling and Prediction of Large-Scale Temporal Variation in Underwater Acoustic Channels," in Proc. of MTS/IEEE OCEANS Conference, Shanghai, China, Apr. 10-13, 2016.
- [25] W. Sun, C. Wang, Z.-H. Wang, and M. Song, "Experimental Comparison Between Under-Ice and Open-Water Acoustic Channels," in Proc. of the ACM International Workshop on Underwater Networks (WUWNet), Washington DC, Oct. 22-24, 2015.
- [26] Z. Wang, L. Wei , X. Lu , Z. Peng and J-H Cui," Towards power efficient deployment for underwater sensing systems", 2015 IEEE 12th International Conference on Networking, Sensing and Control (ICNSC), Taipei, Taiwan, April 9-11, 2015