A REPORT ON

REVIEW OF UNDERWATER ACOUSTIC SENSOR NETWORK APPLICATIONS

Submitted in fulfilment of the course
ECE F266: STUDY PROJECT
Under the guidance of Sarang C. Dhongdi

SHUBHAM AJAY AGRAWAL
2017AAPS0363G
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI K. K. Birla Goa Campus

TABLE OF CONTENTS

1. UnetSIM

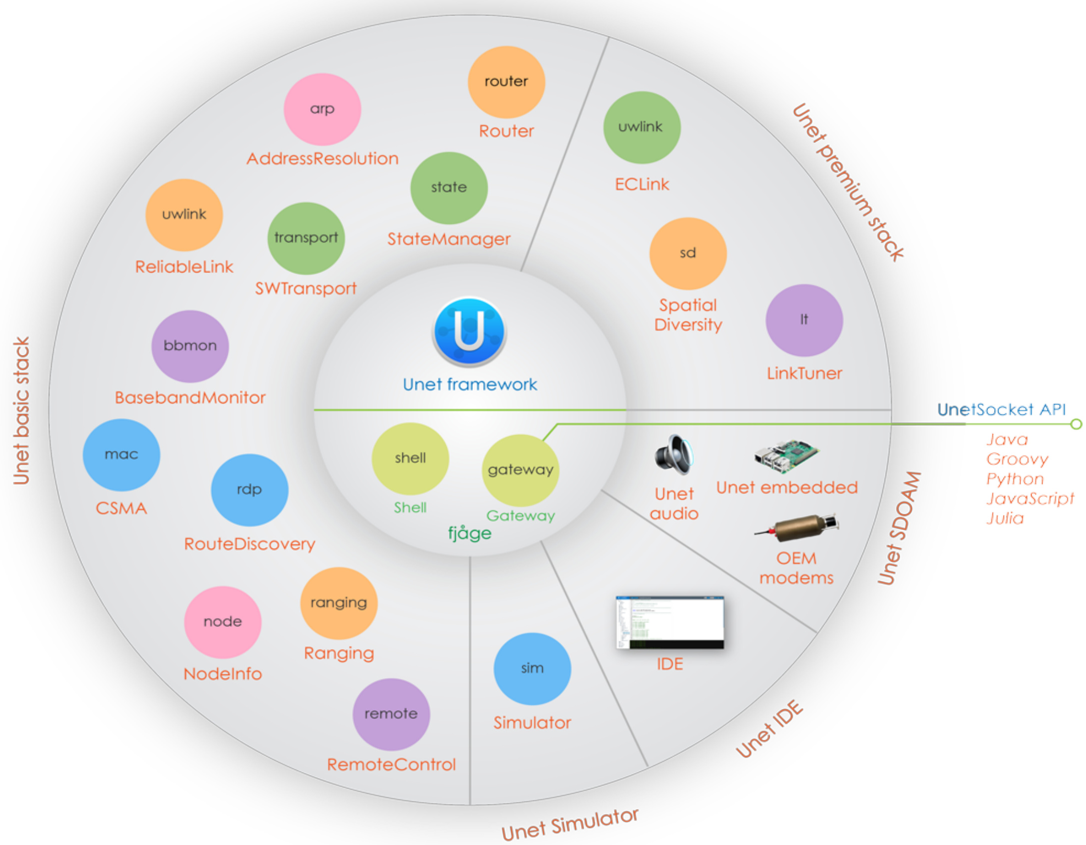
1.1 Introduction

The UnetSim underwater network simulator enables a network engineer to simulate an underwater network on a single computer. It allows UnetStack agents and protocols to be simulated in realistic channel conditions, with minimum effort.

It has two types of simulation:

1. Discrete-Event Simulation Mode: Unet can be used on computers to simulate underwater networks and test protocol performance

2. Real-time Simulation Mode: In this mode, it can be used to interactively debug protocol implementations, and test deployment scenarios prior to an experiment, which once tested can be copied to an underwater modem and deployed in the field.



1.2 UnetStack

Unet nodes run the UnetStack software that allows us to effectively communicate over all types of links using a common Application Programming Interface (API). UnetStack API bindings are available for several languages including Java, Groovy, Python, Julia, C, Javascript, etc.

The Unet stack has an agent-based network stack which is a collection of many agents that have defined capabilities, parameters and provide services likeDifferent agents offer different services as described below:

1. **Node Information**: They manage and maintain node's attributes
2. **Address Regulation**: Are used to manage address requests
3. **Datagram**: They support transmission and reception of messages
4. **Physical**: They are used to send transmission frames
5. **Ranging**: They provide the pair of nodes with time synchronisation
6. **Baseband**: They are used to send recordings of signals
7. **Link**: They provide the nodes with single-hop communication
8. **Medium Access Control (MAC)**: This is used to make the system more reliable in terms of sending acknowledgement messages
9. **Routing**: They provide nodes with multi-hop communication
10. **Route Maintenance**: They generate routing notifications to keep a log of routes in terms of routing tables
11. **Transport**: They provides services such as end to end reliability for large datagrams
12. **Remote Access**: Agents having remote access provide control over nodes
13. **State Persistence**: They, in a way, save the logs or scripts of the present state

The framework thus has good separation of services between different agents; although allowing sharing of information and behaviour negotiation between agents. The agents can also provide additional services that can be used by the other agents as well, which makes the stack extensible and customisable.

In the upcoming series of tasks, we initiate every article by theoretical discussion of the topic and then discussing the simulations and mentioning the conditions assumed for the simulations followed by discussing the output.

2. CHANNEL MODELLING

2.1 Theory

Power level of the nodes refers to the transmission power which the node can provide for sending data. This is the power a packet carries with itself, so as to communicate with a node at a particular distance. The packet while travelling through the medium loses power. The losses are directly proportional to the distance the packet has to travel. Loses are also proportional to the noise level of the medium. A packet needs to be sent with enough power so as to reach the node with an adequate power level.
The channel model used in our simulations is Basic Acoustic Channel. Theoretically, the power level is calculated using the following formula:

c = 1448.96 + 4.591*T - 0.0530*T^2 + 0.0002374*T^3 + 1.340*S'+ 0.0163*d' + 1.675*10^(-7)*d'^2 - 0.01025*T*S' -7.139 * 10^(-13)*T*d'^3;

TL = $10\alpha\log_{10}(R) + (0.0186*(S*fT*f'^2/(fT^2 + f'^2)) + 0.0268 * (f'^2/fT))* 1.0936 * 10^{-3}R$

NL = $N_0 + 10\log_{10}B$

R is the distance between communicating nodes

$S' = S - 35$
$d' = d/2$
$f' = f/1000$
$fT = 21.9 * 10^{(6-1520/(T+273))}$

Total signal-to-noise ratio is given by, SNR = SL − TL − NL , SL is the transmission power level of the source.

Where,
Pt = Source Power Level in dB
Pt' = 10^(Pt/20)
SL = 10log10((10^12)*1000*1500*Pt'/(4*pi))

Also, Received Power Level is given by, RL = SL − TL

Here,
c = sound speed
TL = transmission loss
NL = noise level
f = frequency
B = bandwidth
T = temperature of water in ºC
S = salinity in ppt
α = spreading loss factor

The conditions for BasicAcoustic Channel taken are
carrierFrequency: 25.kHz
bandwidth: 4096.Hz
spreading: 2
temperature: 25.C
salinity: 35.ppt
noiseLevel: 60.dB
waterDepth: 1120.m

2.2 Results

For the given conditions we get,

c = 1543.49 ms^{-1} (approximately for fresh water)

NL= 96.12 dB

The TL varies with the distance between nodes as-

Distance(m)	TL(dB)	Distance(m)	TL(dB)
100	40.26	1100	63.70
200	46.54	1200	64.72
300	50.32	1300	65.67
400	53.08	1400	66.58
500	55.28	1500	67.44
600	57.13	1600	68.26

700	58.73	1700	69.05
800	60.15	1800	69.81
900	61.43	1900	70.54
1000	62.61	2000	71.25

As seen from the above table, the transmission loss increases with distance. As the losses increase, source power level should be increased for a packet to be delivered to longer distances. Also the calculation of speed(1543.49 ms^{-1}) is close to the theoretical speed of acoustic waves in water(1500 ms^{-1}).

3. Distance - Power Level Relation

3.1 Theory

The power level with which a node sends a packet should be enough for the packet to be delivered to the receiving node through the lossy medium. The transmission loss is directly proportional to the distance the packet has to travel between the two nodes. Here we are trying to find the minimum power level with which a packet needs to be sent for its successful transmission.

The channel model used in our simulations is Basic Acoustic Channel.

3.2 Simulation

Here, we attempt to carry different simulations and check the minimum power level for which a node can transmit data to another node at a specified distance. We have done this for distances from 100m to 2000m. We found the optimal power level by calculating the loss incurred in each simulation.

Simulation Conditions:

```
channel = [
  model:              BasicAcousticChannel,     //by default
  carrierFrequency:     25.kHz,
  bandwidth:          4096.Hz,
  spreading:          2,        //spherical or cylindrical (spreading loss factor)
  temperature:        25.C,
  salinity:           35.ppt,
  noiseLevel:         60.dB,   //PSD of ambient noise
  waterDepth:         1120.m  // 20m on net
]
```

Node A

Location = [0.m, 0.m, -100.m]
Address = 1
Static Deployment

Node B

Location = [x.m, 0.m, -100.m]
Address = 2
Static Deployment

Variation in x = 100m, 200m, 300m,.....2000 m.
Variation in power = -5dB, -6dB,.......-40dB
Simulation Time = 2 hrs

3.3 Results

For each simulation, the average loss is calculated by the formula:

loss = trace.txCount ? 100*trace.dropCount/trace.txCount : 0

Distance 500

Power(-dB)	TX Count	RX Count	Loss
--------	--------	--------	------
8	15	15	0.00000
9	15	15	0.00000
10	15	15	0.00000
11	15	15	0.00000
12	15	15	0.00000
13	15	15	0.00000
14	15	15	0.00000
15	15	15	0.00000
16	15	15	0.00000
17	15	15	0.00000
18	15	15	0.00000
19	15	15	0.00000
20	15	15	0.00000
21	15	15	0.00000
22	15	13	13.33333
23	15	14	6.66667
24	15	9	40.00000
25	15	9	40.00000
26	15	2	86.66666
27	15	0	100.00000
28	15	0	100.00000
29	15	0	100.00000
30	15	0	100.00000

SIMULATION RESULT FOR 500M SEPARATION

It can be seen from the above simulation that initially for high power levels all packets were delivered successfully. Then as power level decreases to a certain range, there is a gradual decay in the number of packets successfully delivered. No packets are delivered after decreasing power level crosses a certain value.

So we must set the power level of the node to a value such that all packets are delivered successfully over the distance it needs to travel. Also the power level should not be too high that it reaches the nodes not meant to receive the data.

By simulating the above experiment for different distances, we find the following optimum power levels-

Distance(m)	Optimum PL(-dB)	Distance(m)	Optimum PL(-dB)
100	**35**	**1100**	**8**
200	**30**	**1200**	**8**
300	**26**	**1300**	**8**
400	**23**	**1400**	**5**
500	**21**	**1500**	**7**
600	**20**	**1600**	**5**
700	**17**	**1700**	**3**
800	**14**	**1800**	**3**
900	**14**	**1900**	**1**
1000	**11**	**2000**	**1**

As seen from the above table, the power level needed is more for travelling longer distances.

4. Time Division Multiplexing

4.1 Theory

Time Division Multiplexing is a way to divide a signal into segments, which are transmitted and received in a very short period of time. Hence, it requires the nodes to be timed as when to send signal based on the factors affecting the total time (which includes propagation time, transmission time, and processing time)

Total time, t = Transmission Time + Propagation Time
Total time, t =(Distance / Sound Speed) + (Frame Length / Data Rate)....(i)

4.2 Simulation

In this simulation, six nodes are deployed at a depth 1000m and one more node is deployed at sea level. Five of nodes at 1000m depth send data to the sixth node at 1000m depth using time division multiplexing. The sixth node sends the collected data to the node at sea level. The optimal transmission power levels are set for each node according to the distance the data needs to be transmitted. The time, which the node takes to transmit data to a particular for the given data rate, sound speed and frame length is calculated using formula (i).
Frame Length = 64 bits
Data Rate = 2400 bps

Simulation Conditions:
Conditions:
channel = [

model:	BasicAcousticChannel
carrierFrequency:	25.kHz,
bandwidth:	4096.Hz,
spreading:	2,
temperature:	25.C,
salinity:	35.ppt,
noiseLevel:	60.dB,
waterDepth:	1120.m

]

modem.dataRate = [2400, 2400].bps
modem.frameLength = [8, 8].bytes

Nodes:

Node A - Base Station
Location = [0.m, 0.m, 0.m]
Address = 10
Static Deployment

Node B
Location = [0, 0, -1000.m]
Address = 21
Data collected before transmission = [51,52,53,54,55]
Data transmitted at time, t = 0.920 sec
Transmission Power Level = -13 dB
Static Deployment

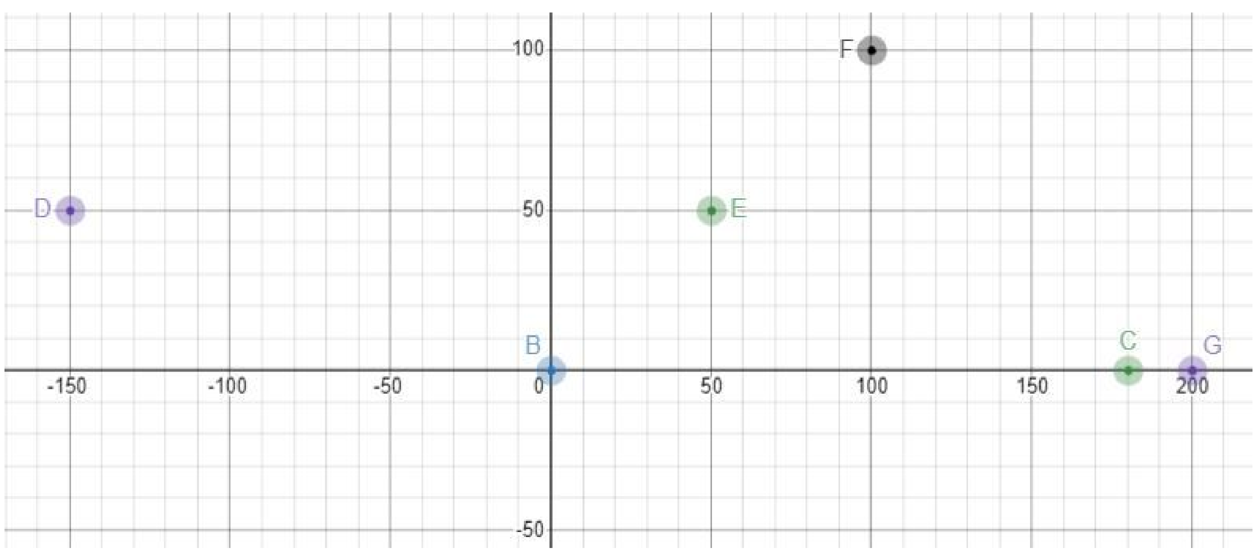
Node C
Location = [180.m, 0, -1000.m]
Address = 31
Data collected before transmission = [51]
Data transmitted at time, t = 0.064 sec
Transmission Power Level = -30 dB
Static Deployment

Node D
Location = [-150.m, 50.m, -1000.m]
Address = 32
Data Contained = [52]
Data transmitted at time, t = 0.249 sec
Transmission Power Level = -32 dB
Static Deployment

Node E
Location = [50.m, 50.m, -1000.m]
Address = 33
Data Contained = [53]
Data transmitted at time, t = 0.419 sec
Transmission Power Level = -35 dB
Static Deployment

Node F
Location = [100.m, 100.m, -1000.m]
Address = 34
Data Contained = [54]
Data transmitted at time, t = 0.561 sec
Transmission Power Level = -32 dB
Static Deployment

Node G
Location = [200.m, 0, -1000.m]
Address = 35
Data Contained = [55]
Data transmitted at time, t = 0.721 sec
Transmission Power Level = -30 dB
Static Deployment



X and Y coordinate plot of nodes B to G at depth 1000m

4.3 Results

Expected Result:
The time, which the node takes to transmit data to a particular node for the given data rate, sound speed and frame length is calculated using formula (i).
Frame Length = 64 bits
Data Rate = 2400 bps

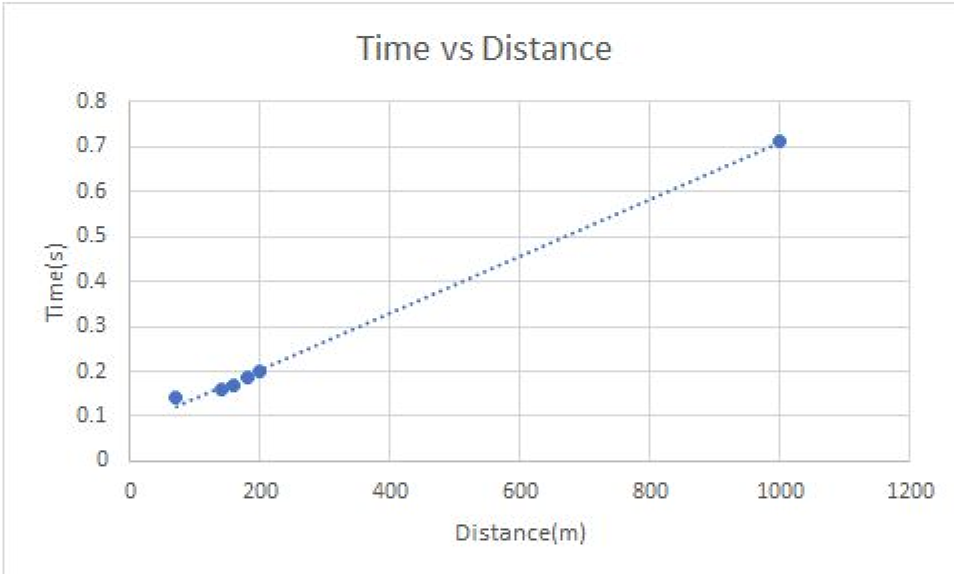
The screenshot below shows the detailed log of data transmission, and first column of every row mentions the time in ms, at which the action is performed.

```
64|INFO|Test_Agent@74:println|data sent by 31 to 21
181|INFO|Test_Agent@94:println|message processed in address:21
249|INFO|Test_Agent@94:println|message processed in address:21
249|INFO|Test_Agent@94:println|DATA Received by 21 from 31, DATA VALUE is [51]
249|INFO|Test_Agent@94:println|now total value is [51]
249|INFO|Test_Agent@78:println|data sent by 32 to 21
351|INFO|Test_Agent@94:println|message processed in address:21
419|INFO|Test_Agent@94:println|message processed in address:21
419|INFO|Test_Agent@94:println|DATA Received by 21 from 32, DATA VALUE is [52]
419|INFO|Test_Agent@94:println|now total value is [51, 52]
419|INFO|Test_Agent@82:println|data sent by 33 to 21
493|INFO|Test_Agent@94:println|message processed in address:21
561|INFO|Test_Agent@94:println|message processed in address:21
561|INFO|Test_Agent@94:println|DATA Received by 21 from 33, DATA VALUE is [53]
561|INFO|Test_Agent@94:println|now total value is [51, 52, 53]
561|INFO|Test_Agent@86:println|data sent by 34 to 21
653|INFO|Test_Agent@94:println|message processed in address:21
721|INFO|Test_Agent@94:println|message processed in address:21
721|INFO|Test_Agent@94:println|DATA Received by 21 from 34, DATA VALUE is [54]
721|INFO|Test_Agent@94:println|now total value is [51, 52, 53, 54]
721|INFO|Test_Agent@90:println|data sent by 35 to 21
851|INFO|Test_Agent@94:println|message processed in address:21
919|INFO|Test_Agent@94:println|message processed in address:21
919|INFO|Test_Agent@94:println|DATA Received by 21 from 35, DATA VALUE is [55]
919|INFO|Test_Agent@94:println|now total value is [51, 52, 53, 54, 55]
920|INFO|Test_Agent@94:println|[51, 52, 53, 54, 55] in startup in addr: 21
920|INFO|Test_Agent@94:println|data sent by 21 to 10
920|INFO|Test_Agent@94:println|message processed in address:21
1636|INFO|Test_Agent@98:println|DATA Received by Base Station with address: 10 , DATA VALUE is: [51, 52, 53, 54, 55]
3600000|INFO|Script1@100:call|Stopping simulation...
```

From the above log file we get the time a packet takes to be delivered,

Nodes	**Distance(m)**	**Time taken for Transfer(s)**
C to B	180.00	0.19
D to B	158.11	0.17
E to B	70.71	0.14
F to B	141.42	0.16
G to B	200.00	0.20
B to A	1000.00	0.71

We plot a time vs distance graph from the above table,



The Best Fit Line for the above data gives the equation -
Time = 0.00063*Distance + 0.07666 …..(ii)

Taking the inverse of the slope gives a value **1587.30** which is close to the speed of acoustic waves in water.

Each node send 1 frame which takes time 64/2400= 0.02666 which is included in the constant part of the equation (ii).

The extra time taken **(0.05s)** for packets to be received at the receiving end includes the processing time. Hence, the values are closer to the expected values, and using TDM we are able to send data without any packet drops due to bad frames or not detected and the whole process takes 1.636s for the base station to receive the data at the receiving end as expected.

5. Mobility

5.1 Theory

Mobility in wireless networks basically refers to a node, Mobile Node (MN), or sometimes a subnet, changing its point of attachment to the network while its communication to the network remains uninterrupted. Mobility reduces energy consumption and provides better monitoring of ongoing events in sensed fields.

The mobility of the nodes is set to false to indicate that the nodes are static and it set to true if the nodes are mobile.

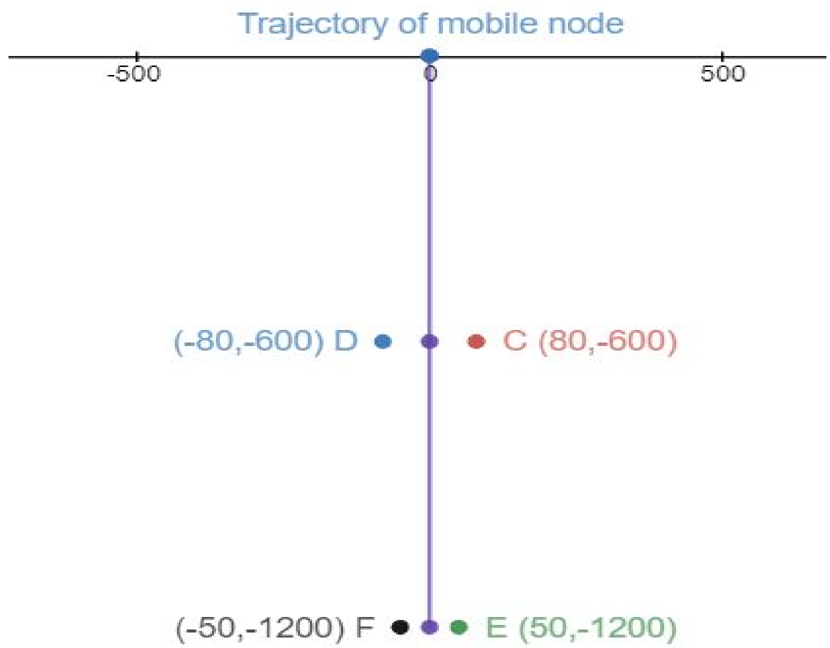
Parameters involved in Mobility:
speed — speed in m/s, if mobile node
heading — heading in degrees, 0 is North, measured clockwise, if mobile node
turnRate — turn rate in degrees/s, measured clockwise, if mobile node
diveRate — dive rate in m/s, if mobile node

5.2 Simulation 1

Program 1 : A mobile node with linear motion along the depth communicates with its neighbouring nodes as it moves.

```
node 'C', address: 31, location: [80.m, 0, -600.m]

node 'D', address: 32, location: [-80.m, 0, -600.m]

node 'E', address: 33, location: [50.m, 0, -1200.m]

node 'F', address: 34, location: [-50.m, 0, -1200.m]

node 'B', address: 21, location: [0, 0, 0], mobility: true
        motionModel = [
            [duration: 3.minutes,  diveRate: 10.mps],
            [duration: 3.minutes,  diveRate: -10.mps],
            [time: 6.minutes, turnRate: 0.dps, diveRate: 0.mps],
        ]
```

The power levels are set to a level at which communication is possible only when the nodes are close to each other.



Side view of the mobility model-1

Here we have given node 'B' a motion model where it first dives downwards for 3 minutes with a speed of 10ms-1. Then it retraces its path upwards in 3 minutes with the same speed and then rests at the sea level.

5.3 Results

In this simulation, as the mobile node reaches 600m depth, it communicates with nodes C and D. Similarly as it moves to 1200m depth, it communicates with the nodes E and F.

```
59000|INFO|Test_Agent@16:println|data sent by 31 to 21
59052|INFO|Test_Agent@32:println|message processed in address:21
59120|INFO|Test_Agent@32:println|message processed in address:21
59120|INFO|Test_Agent@32:println|DATA Received by 21 from 31, DATA VALUE is [51]
59120|INFO|Test_Agent@32:println|now total value is [51]
61000|INFO|Test_Agent@20:println|data sent by 32 to 21
61052|INFO|Test_Agent@32:println|message processed in address:21
61120|INFO|Test_Agent@32:println|message processed in address:21
61120|INFO|Test_Agent@32:println|DATA Received by 21 from 32, DATA VALUE is [52]
61120|INFO|Test_Agent@32:println|now total value is [51, 52]
119000|INFO|Test_Agent@24:println|data sent by 33 to 21
119033|INFO|Test_Agent@32:println|message processed in address:21
119101|INFO|Test_Agent@32:println|message processed in address:21
119101|INFO|Test_Agent@32:println|DATA Received by 21 from 33, DATA VALUE is [53]
119101|INFO|Test_Agent@32:println|now total value is [51, 52, 53]
121000|INFO|Test_Agent@28:println|data sent by 34 to 21
121033|INFO|Test_Agent@32:println|message processed in address:21
121101|INFO|Test_Agent@32:println|message processed in address:21
121101|INFO|Test_Agent@32:println|DATA Received by 21 from 34, DATA VALUE is [54]
121101|INFO|Test_Agent@32:println|now total value is [51, 52, 53, 54]
180000|INFO|<B>@33:invoke|Motion update for B: [duration:180, diveRate:-10]
360000|INFO|<B>@33:invoke|Motion update for B: [time:360, turnRate:0, diveRate:0]
3600000|INFO|Script1@34:call|Stopping simulation...
3600000|INFO|org.arl.unet.sim.SimulationContainer@34:shutdown|Initiating shutdown...
3600000|INFO|org.arl.unet.sim.SimulationContainer@34:shutdown|All agents have shutdown
```

Result file for simulation 1

From the above log file we conclude that the mobile node successfully collected data from all the other nodes. We can also see the motion updates of the mobile node at 180s and 360s. The node is successfully following the motion model.

5.4 Simulation 2

Program 2 : A mobile node at a certain depth moves in a horizontal circular trajectory. 4 nodes are located a different depth level communicate with the mobile node when it is closest to them.

node 'C', address: 31, location: [80.m, 80.m, -50.m

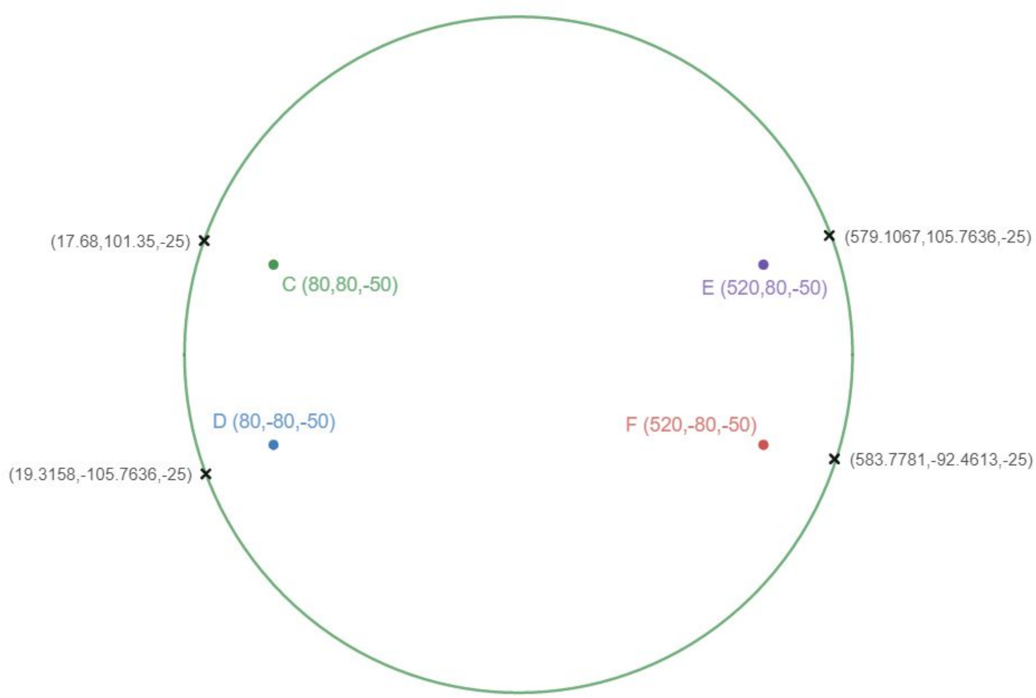
node 'D', address: 32, location: [80.m, -80.m, -50.m]

node 'E', address: 33, location: [520.m, 80.m, -50.m]

node 'F', address: 34, location: [520.m, -80.m, -50.m]

node 'B', address: 21, location: [0, 0, -25.m], mobility: true
motionModel = [speed: 47.mps, turnRate: 9.dps]

The power levels are set to a level at which communication is possible only when the nodes are close to each other.



Top view of the mobility model-2

Here we have given node ‘B’ a motion model where it turns clockwise with a rate of 9 degrees per second and has a speed of 47mps. So the overall trajectory becomes a circle of radius 300m which it covers in 40s.

5.5 Results

In this simulation, the node communicates with the nodes C, E, F and D at positions (17.68,101.35), (579.1,105.7), (583.7,-92.46) and (19.3,-105.76) respectively as shown in the figure.

```
2200|INFO|Test_Agent@16:println|data sent by 31 to 21
2246|INFO|Test_Agent@32:println|message processed in address:21
2314|INFO|Test_Agent@32:println|message processed in address:21
2314|INFO|Test_Agent@32:println|DATA Received by 21 from 31, DATA VALUE is [51]
2314|INFO|Test_Agent@32:println|now total value is [51]
17700|INFO|Test_Agent@28:println|data sent by 34 to 21
17827|INFO|Test_Agent@32:println|message processed in address:21
17895|INFO|Test_Agent@32:println|message processed in address:21
17895|INFO|Test_Agent@32:println|DATA Received by 21 from 34, DATA VALUE is [54]
17895|INFO|Test_Agent@32:println|now total value is [51, 54]
22000|INFO|Test_Agent@24:println|data sent by 33 to 21
22120|INFO|Test_Agent@32:println|message processed in address:21
22188|INFO|Test_Agent@32:println|message processed in address:21
22188|INFO|Test_Agent@32:println|DATA Received by 21 from 33, DATA VALUE is [53]
22188|INFO|Test_Agent@32:println|now total value is [51, 54, 53]
37700|INFO|Test_Agent@20:println|data sent by 32 to 21
37746|INFO|Test_Agent@32:println|message processed in address:21
37814|INFO|Test_Agent@32:println|message processed in address:21
37814|INFO|Test_Agent@32:println|DATA Received by 21 from 32, DATA VALUE is [52]
37814|INFO|Test_Agent@32:println|now total value is [51, 54, 53, 52]
3600000|INFO|Script1@34:call|Stopping simulation...
3600000|INFO|org.arl.unet.sim.SimulationContainer@34:shutdown|Initiating shutdown...
3600000|INFO|org.arl.unet.sim.SimulationContainer@34:shutdown|All agents have shutdown
```

Result file for simulation 2

From the above log file we conclude that the mobile node successfully collected data from all the other nodes. The mobile node communicated successfully with the static nodes when it was closest to them.

Having a mobile node saves a lot of transmission power, as the power required for the static nodes to communicate would be a lot more.

6. Bibliography

List of Reference Book(s), Journal(s), Other Authentic Document(s)/Source(s):

1. Unet (www.unetstack.net)
2. fjåge 1.6.2 documentation(https://fjage.readthedocs.io/)
3. https://ns2blogger.blogspot.com/
4. Salvador Climent, Antonio Sanchez, Juan Vicente Capella, Nirvana Meratnia 2 and Juan Jose Serrano 1, (2014), Underwater Acoustic Wireless Sensor Networks: Advances and Future Trends in Physical, MAC and Routing Layers.
5. Mandar Chitre, Shiraz Shahabudeen, Milica Stojanovic, (2008), Underwater Acoustic Communications and Networking: Recent Advances and Future Challenges.
6. http://docs.groovy-lang.org/